



TREBALL DE FI DE GRAU

GRAU D'ENGINYERIA INFORMÀTICA

Facultat de Matemàtiques

Universitat de Barcelona

ESTUDI DE METODOLOGIES ÀGILS PER AL DESENVOLUPAMENT D'UNA APLICACIÓ ANDROID

CAROLINA BONNIN ABAD

Director: Oriol Pujol Vila

Realitzat a: Departament de matemàtica aplicada i anàlisi.

Data: Dilluns 22 de juny de 2015.

ABSTRACT

This report contains a theoretical and applied study of three main agile project management methodologies: Scrum, Kanban and Lean. It contains a detailed study of the main features of these methodologies and present their historical background, as well as actors, artifacts and ceremonies involved in each methodology.

At the same time of the theoretical study, this report presents the adaptation of these methodologies to be applied in the development of an Android application, and the conclusions drawn after applying each one, the negative and positive results obtained both individually for each methodology and globally once the application development are drawn.

RESUM

La present memòria conté un estudi teòric-aplicat de tres de les principals metodologies de gestió de projectes àgils: Scrum, Kanban i Lean. S'estudiaran les principals característiques d'aquestes metodologies i es presentaran els seus antecedents històrics, així com els actors, artefactes i cerimònies implicades en cada metodologia. Paral·lelament a l'estudi teòric, també es presentarà l'adaptació de les tres metodologies per a ser aplicades en el desenvolupament d'una aplicació Android, les conclusions extretes després de cadascuna d'elles, els punts negatius i positius i els resultats obtinguts, tant a nivell individual de cada metodologia com a nivell global un cop finalitzat el desenvolupament de l'aplicació.

AGRAÏMENTS

M'agradaria donar les gràcies, primerament, al meu tutor i cap d'estudis, Oriol Pujol per la seva paciència, perseverança, bon humor i bons consells durant la realització d'aquest projecte i el transcurs de la carrera; si no fos per ell, potser no hagués arribat a realitzar aquest treball.

També donar-li les gràcies al meu professor Eloi Puertas per haver-se preocupat de l'evolució d'aquest projecte, haver sigut un gran professor i per exigir sempre el màxim de mi en les seves assignatures.

Agrair a tot el professorat que m'ha traspassat una part dels seus coneixements al llarg de la carrera, per la seva vocació i coneixement en les matèries impartides.

Finalment, gràcies a tots els meus companys que han cursat els quatre anys de carrera amb mi, especialment a A. Vilalta per haver sigut i seguir sent un gran amic, suport, psicòleg, company d'estudi, professor no titular i una gran companyia durant tot aquest temps, i per tots els projectes que ens esperen, a la meva família per haver tingut la paciència d'escoltar-me, fer-me costat i haver fet aquest somni realitat, a D. Sopov per haver sigut el meu suport i un referent d'exigència, organització i estima durant aquest curs i finalment a N.M. Montes per haver estat sempre al meu costat, sent un referent de força de voluntat, amicitat i optimisme.

Moltes gràcies a tots.

ÍNDEX

1. Introducció i motivacions.....	6
Estructura de la memòria.....	6
Introducció.....	6
2. Objectius.....	8
3. Estudi de les metodologies.....	9
3.1 Agile.....	9
3.2. Scrum.....	18
3.3. Kanban.....	25
3.2 Lean.....	32
4. Posada en pràctica. Desenvolupament.....	38
4.1 Scrum.....	41
4.2 Kanban.....	61
4.3 Lean.....	76
5. Resultats finals de l'aplicació.....	92
6. Conclusions finals i comparativa.....	96
7. Referències i bibliografia.....	98
Annex número 1.....	100

1. INTRODUCCIÓ I MOTIVACIONS.

Estructura de la memòria

La present memòria, iniciada per una introducció on es contextualitza el projecte, s'estructura en dues parts principals, la primera part consta d'un estudi teòric de tres metodologies punteres en la gestió de projectes àgils: Scrum, Kanban i Lean, incloent també l'estudi de la filosofia Agile per a la gestió de projectes en primera instància. Dins d'aquest estudi teòric trobarem exposades les principals característiques de cada metodologia, els seus antecedents històrics i els principals actors, artefactes i cerimònies particulars.

En la segona part de la memòria, es presentarà la part pràctica del projecte, l'aplicació de les tres metodologies per al desenvolupament d'una aplicació Android. Per a l'aplicació de cada metodologia s'exposarà l'adaptació feta per tal d'aplicar-les en el projecte, els resultats obtinguts un cop finalitzat el desenvolupament i les conclusions i punts positius i negatius extrets després de la seva aplicació.

Finalment trobarem una conclusió referida als objectius principals del projecte i la bibliografia enumerada.

Introducció

La gestió de projectes i l'aplicació de metodologies *Agile* per al desenvolupament de software és un tendència que cada cop més, va prenent força en l'organització de les empreses per obtenir millors resultats.

Els models de gestió de projectes consisteixen en una sèrie de pràctiques, mètodes, principis, tècniques i eines que tenen com a objectiu la planificació i seguiment del treball realitzat, amb l'objectiu de millorar el rendiment de l'equip, prioritzar al client i optimitzar el temps d'entrega dels projectes, sempre reduint costos als llarg del procés.

Avui en dia, dins del mon *Agile*, algunes de les metodologies que creen més tendència son Scrum, Kanban i Lean. Algunes més conegudes que altres, són completament compatibles entre

elles, i el seu estudi per separat ens dona les eines per poder-les aplicar en equips de producció de software.

Paral·lelament, el desenvolupament d'aplicacions Android pren força en el mercat i en el nostre dia a dia; parlem de la necessitat de tenir eines al nostre mòbil o tablet que ens facilitin les tasques més rutinàries i que ens mantinguin completament connectats i informats amb la resta del món.

Aquest treball té com a principal objectiu l'estudi teòric-aplicat de les tres metodologies abans esmentades de gestió de projectes àgils, Scrum, Kanban i Lean.

Per tal de poder aplicar aquestes metodologies en un context pràctic, s'ha definit l'objectiu de crear una aplicació Android d'acompanyament per fer tours a peu. En particular, ens permet seleccionar una ruta a seguir sobre un mapa, previsualitzar el trajecte i poder escoltar i experimentar aquest trajecte mitjançant la música de les persones del nostre voltant.

El desenvolupament d'aquesta aplicació s'ha dividit en tres parts, en cada part s'ha aplicat una de les tres metodologies de gestió de projectes, podent realitzar un anàlisi a nivell teòric i pràctic, obtenint, com a resultat d'un treball organitzat i estructurat, una sèrie de dades analitzables, així com una aplicació completament usable i testejada.

D'aquesta manera, les conclusions extretes després de realitzar el treball, tenen un fonament pràctic, gràcies a l'aplicació de les metodologies en un projecte real de desenvolupament de software, i teòric, gràcies a tota la informació obtinguda i estudiada sobre models de gestió.

La principal motivació d'aquest projecte ha sigut, per tant, l'oportunitat de poder aplicar, entendre i estudiar tres metodologies punteres de gestió de projectes, obtenint com a resultat una aplicació Android.

Tots aquests punts es presentaran i analitzaran al llarg d'aquesta memòria, posant l'accent en l'estudi de les diferents metodologies aplicades i els resultats obtinguts.

2. OBJECTIUS.

Els objectius d'aquest projecte són:

- **Anàlisi de tres metodologies de gestió de projectes**
 - a. Les tres metodologies escollides per a l'estudi són Scrum, Kanban i Lean.
- **Identificació i posada en pràctica**
 - a. Com aplicar les metodologies a les circumstàncies concretes del projecte.
- **Aplicació d'aquestes metodologies en el desenvolupament d'una aplicació Android.**
 - a. Divisió de l'aplicació en tres fases de desenvolupament per tal d'aplicar un model de gestió diferent en cada fase.

3. ESTUDI DE LES METODOLOGIES.

3.1 Agile

Quan parlem d'Agile, no només ens referim a un conjunt de metodologies pel desenvolupament de projectes, sinó també a una nova filosofia organitzativa i de treball dins de les empreses.

3.1.1 Antecedents històrics

La idea del desenvolupament en cicles curts, iteratius i incrementals, en un inici anomenat model iteratiu incremental, principal característica de les metodologies àgils, data del 1939, anys abans de que Winston Royce¹ definís els primers esbossos de la metodologia en cascada. Durant dècades, els models iteratius incrementals van formar part de molts projectes, però no hi havia cap definició concreta de la seva aplicació.

No va ésser fins als anys '90 que el reconeixement públic de les metodologies de desenvolupament emmarcades en un cicle iteratiu incremental va prendre força, la definició moderna de desenvolupament Agile va començar definir-se, especificant les característiques d'aplicació del model iteratiu incremental, desenvolupat com a contraposició del model en cascada, que predominava durant els anys seixanta, amb la principal filosofia de programar i testejar al final. El problema era que aquest model no s'adaptava a la realitat del processos de desenvolupament de software.

Els primers esbossos de la definició d'Agile van ser plantejats per James Martin², i es van donar a conèixer a la comunitat d'Enginyeria de Software amb el nom de RAD (Rapid Application Development). Aquest nou enfoc es basava en una planificació mínima i un prototipat ràpid amb entorns de desenvolupament altament productius, entregues ràpides al client i estalvi de desenvolupament.

¹ Winston Royce: científic de computadors Americà, director del Centre de Tecnologia de Software Lockheed, Austin, Texas. Creador del model en cascada.

² James Martin (1933 – 2013): consultor de IT i escriptor britànic. Conegut per la creació del "*Rapid Application Development*".

Tot i sent RAD el precedent de la idea àgil, el veritable inici d'aquesta metodologia es dona amb la creació d'XP (eXtreme Programming), metodologia ideada per Kent Beck ³ i Ward Cunningham⁴.

Les metodologies àgils, tal i com les coneixem avui en dia s'anomenaven metodologies lleugeres, i la major part dels programadors no ho consideraven exactament una metodologia de gestió, ja que no era tant estricta ni organitzada com les vistes anteriorment, sinó més aviat com un procés intuïtiu dins de l'empresa o compartit entre membres de l'equip.

L'any 1996 Kent Beck va aplicar tots els coneixements recopilats durant anys per salvar el projecte Chrysler Comprehensive Compensation (C3), donant origen a XP i iniciant el moviment de les metodologies àgils.

L'any 2001, després d'una reunió a Utah, Estats Units, neix formalment el terme Agile aplicat al desenvolupament de software.

En aquesta reunió van col·laborar disset experts de la indústria del software amb l'objectiu de definir els valors i principis que havien de permetre als equips desenvolupar software de manera més eficient, responent als possibles canvis i variacions que pot sofrir el projecte durant el seu desenvolupament. Tots els valors definits eren una especificació més concreta de totes les aplicacions que s'havien fet, sense cap patró clar, del model iteratiu incremental.

Després d'aquesta reunió es crea *The Agile Alliance*, una organització dedicada a promoure els conceptes relacionats amb el desenvolupament àgil de software i ajudar a les empreses a aplicar aquests conceptes, es defineixen els patrons d'aplicació de les metodologies Agile.

La conclusió d'aquesta famosa reunió l'any 2001, va ser el manifest Agile.

³ Kent Beck: enginyer de software d'Estats Units, conegut per ser un dels creadors de la programació extrema (eXtreme Programming).

⁴ Ward Cunningham: programador de patrons per a la programació. Creador del primer lloc de tipus "wiki", el *WikiWikiWeb*. Conegut, també, per ser un dels creadors de XP.

3.1.2. *Manifest Agile*

Estem descobrint maneres millors de desenvolupar software tant per la nostra pròpia experiència com ajudant a tercers. A través d'aquest treball hem après a valorar:

Individus i interaccions sobre processos i eines.

Software funcionant sobre documentació extensiva.

Col·laboració amb el client sobre negociació contractual.

Resposta davant el canvi sobre seguir un pla.

Això és, encara que valorem els elements de la dreta, valorem més els de l'esquerra.

Un cop creat el manifest Agile, com a principal referent, sorgeixen els principis àgils, que enumeren les principals diferències en relació a un model predictiu.

3.1.3. *Principis Agile*

1. La nostra major prioritat és satisfer al client mitjançant l'entrega primerenca i continua de software amb valor.
2. Acceptem que els requisits canvien, inclús en etapes tardanes del desenvolupament. Els processos àgils aprofiten el canvi per proporcionar avantatge competitiva al client.
3. Entreguem software funcional freqüentment, entre dos setmanes i dos mesos, amb preferència al període de temps més curt possible.
4. Els responsables de negoci i els desenvolupadors trebalem junts de forma quotidiana durant tot el projecte.
5. Els projectes es desenvolupen entorn a individus motivats. Cal donar-los l'entorn i el suport que necessiten, i confiar-los l'execució del treball.
6. El mètode més eficient i efectiu de comunicar informació de l'equip de desenvolupament i entre els seus membres és la conversació cara a cara.
7. El software funcionant és la mesura principal de progrés.
8. Els processos Àgils promouen el desenvolupament sostenible. Els promotors, desenvolupadors i usuaris hem de ser capaços de mantenir un ritme constant de forma indefinida.

9. L'atenció continua a l'excel·lència tècnica i al bon disseny millora l'Agilitat.
10. La simplicitat, o l'art de maximitzar la quantitat de treball no realitzar, és essencial.
11. Les millors arquitectures, requisits i dissenys emergeixen d'equips auto-organitzats.
12. A intervals regulars l'equip reflexiona sobre com ser més efectiu per a continuació ajustar i perfeccionar el seu comportament en conseqüència.

3.1.4. Disussió

Agile es basa en el cicle de vida iteratiu incremental, creat com a alternativa a les metodologies predictives, que es basen en cicles de vida clàssics o de model en cascada.

Per tant, per poder entendre correctament el concepte Agile, primer hem de conèixer el cicle de vida iteratiu incremental i el seu contraexemple, és a dir, els models predictius.

Model cascada o predictiu

Les principals característiques del model cascada o predictiu són, una fase de planificació detallada, on es visualitza el resultat del sistema i les estimacions de temps, la revisió i aprovació d'aquest pla abans de que l'equip comenci a treballar, un producte dissenyat al detall des del primer moment, equips amb membres especialitzats que realitzen les seves tasques com a especialistes de l'àrea corresponent, documentació molt detallada un cop acabat el projecte, el pas a test del projecte un cop acabat i un control molt estricte per evitar desviacions en el pla original.

Com es pot observar, es tracta d'un model de gestió molt estricte, on es defineix un pla inicial i pretén no variar-lo fins a la finalització del projecte. El testeig del correcte funcionament del sistema no es realitza en entorns de test fins que està acabat.

El fet de que sigui un model tant estricte ens aporta punts molt positius, com ara que és lògic i metòdic, segueix un pla i manté l'orde tant com pot.

En contra, també ofereix punts negatius, com ara la poca flexibilitat de cara als errors i als canvis, degut a una definició inicial estricta del projecte que difícilment admet modificacions durant el desenvolupament, i la falta de previsió als futurs riscos un cop acabat el desenvolupament.

Divisió del cicle de vida segons el model predictiu.

Tot i que poden haver-hi variacions dins d'aquestes etapes de vida segons cada empresa o enfoc, s'han escollit les definides per Roger Pressman⁵.

Primerament, es comença amb una fase d'anàlisi, que implica l'obtenció d'uns requisits inicials per determinar la viabilitat del sistema i analitzar diferents solucions, l'elaboració dels requisits, la realització dels casos d'ús i el prototipat de pantalles així com la redacció d'informes per especificar el pla de proves.

En segon terme, passem al disseny, que fa referència a la descripció del sistema sense entrar en el detall de la implementació. Es va concretant la solució al problema, es defineix l'arquitectura general del sistema, el model de dades, el model de classes etc.

Seguidament, la codificació, que és la construcció i proves referents al procés. Aquestes proves impliquen tant proves unitàries com proves d'integració i de sistema i la càrrega inicial de dades.

Un cop superades les proves a nivell de codificació, es passa als entorns de preproducció, on es realitzen les proves del desenvolupament realitzat, i on es valida que el sistema fa el que s'espera d'ell. Finalment, quan es superen tots els tests en l'entorn de preproducció es passa a l'entorn de producció.

Finalment, passem al manteniment; un cop el sistema es troba a producció, es realitzen diferents tasques de manteniment que inclouen correccions, evolucions, adaptacions i perfeccionament del sistema.

⁵ Enginyer de software nord-americà, autor i consultor, president de la R.S. Pressman & Associates.

Autor de publicacions com ara:

1982. *Software engineering : a practitioner's approach* (Primera edició).

1988. *Making software engineering happen : a guide for instituting the technology*.

1988. *Software engineering : a beginner's guide*.

Entre altres.

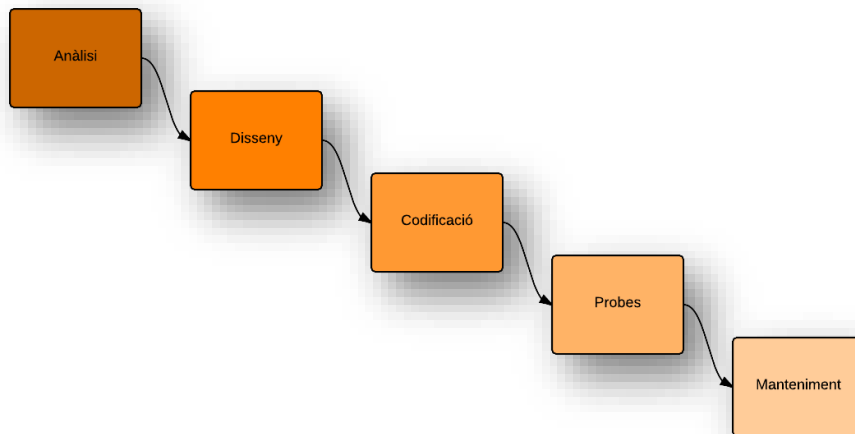


Figura 1. Diagrama del cicle de vida en un model predictiu o cascada.

Com es pot observar a la figura 1, el model en cascada es caracteritza per ordenar seqüencialment les fases de desenvolupament d'un projecte, sense retroalimentació entre les diferents fases. Comença per l'anàlisi, seguit del disseny, codificació, probes i finalment manteniment.

El cicle de vida iteratiu i incremental

Sense cap mena de dubte, una de les principals característiques de qualsevol projecte àgil és l'aplicació d'un cicle de vida iteratiu i incremental, en contraposició al cicle de vida en cascada, vist anteriorment.

S'ha de tenir en compte que el model iteratiu va néixer inclús abans que el model en cascada, cap als anys 60, tot i així, el primer mètode àgil no es va aplicar fins l'any 1994.

En el cicle de vida incremental, el producte es va desenvolupant en iteracions, al final de cada iteració s'obté una versió lliurable del producte, fet que implica que el projecte es desenvolupa per increments, obtenint un feedback constant al final de cada iteració.

En cada increment es realitza una fase d'anàlisi, una de disseny, codificació i probes/acceptació, podent realitzar diverses iteracions simultànies, sempre que aquestes siguin independents entre elles.

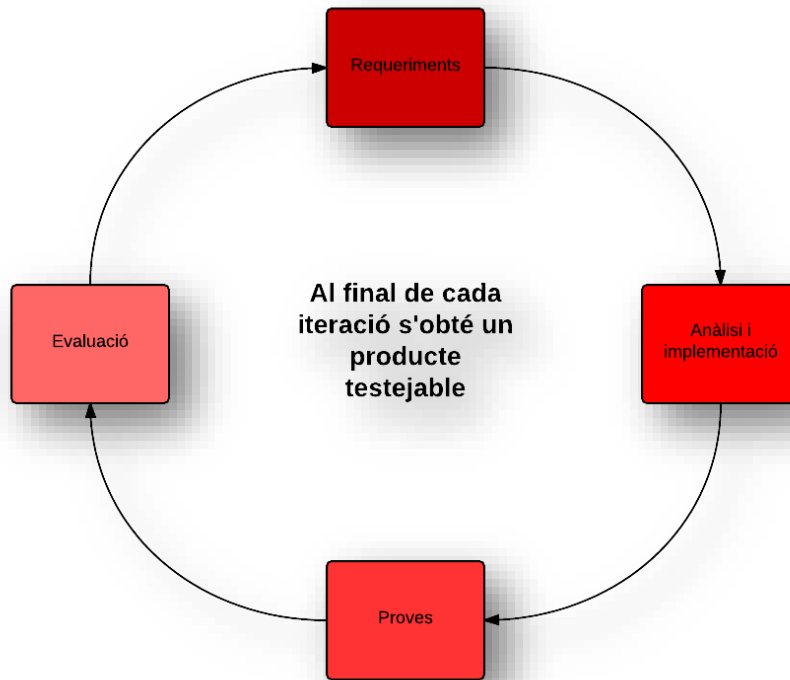


Figura 2. Iteració en el model iteratiu i incremental.

Com podem veure a la figura 2, el model de vida iteratiu i incremental es caracteritza per la seva representació circular, la finalització d'un cicle implica l'inici del següent, de manera que al iniciar el cicle reaprofitem tot el coneixement obtingut en el cicle anterior. Comença per la selecció de requeriments, anàlisi i implementació, fase de proves, avaluació i tornar a començar amb els requeriments, així fins que es considera que el producte està acabat i compleix tots els requeriments.

El model iteratiu i incremental es caracteritza per estar format per equips multi-funcionals auto-gestionats, és a dir, que prenen les decisions de desenvolupament i poden cobrir un ampli ventall de coneixements, per la manca de jerarquies, iteracions ràpides, la construcció incremental del projecte i pel paper fonamental del client durant el desenvolupament.

Les principals avantatges d'aquest model són la rapidesa, la flexibilitat als canvis, la detecció d'errors durant les iteracions que fa que siguin més fàcils corregir, i per tant la reducció de riscos, l'acceptació de modificacions durant el procés de desenvolupament entesos com una avantatge competitiva i la bona visibilitat del projecte mentre es va desenvolupant, que implica un millor manteniment de codi i un procés de desenvolupament simplificat. A part, degut a que els equips són auto-gestionats, també estan més motivats i són més productius, de la mateixa manera que

el client també està més satisfet, ja que obté versions del producte després de cada iteració i participa activament en el procés.

En contraposició, també trobem punts negatius, com ara la poca o nul·la documentació un cop acabat el projecte i la poca visió del resultat futur quan s'inicia el projecte.

Clàssic vs. Iteratiu

A continuació es mostra una taula comparativa entre les dues metodologies, comparant els enfocaments que dona cadascuna segons les diferents situacions.

	MODEL PREDICTIU	MODEL ITERATIU
ENFOC DEL PROJECTE	Aborda el projecte de manera estricta , es basa en l'abast, el cost i el temps .	Planteja el projecte amb un objectiu més ampli , generar software amb el major valor pel client possible.
DEFINICIÓ DE L'ABAST DEL PROJECTE	Definició detallada de l'abast del projecte.	Abast global a l'inici, es va detallant a mida que s'acaba cada iteració .
INDIVIDUS I EINES	No li dona massa importància a la qualitat dels professionals, ja que es basa en la monitorització i el control .	Proposa equips auto-gestionats , prioritza la qualitat dels professionals i els anima a involucrar-se activament en el projecte.
DOCUMENTACIÓ	Planifica detalladament les etapes i les documenta	La documentació queda en segon pla .

COL·LABORACIÓ DEL CLIENT	Col·labora en l'elaboració de l' objectiu i els requeriments . Participa activament durant el desenvolupament, està en contacte amb els desenvolupadors.
RESPOSTA ALS CANVIS	No considera que hi hagin canvis ni modificacions un cop s'ha definit el pla, els objectius i els requeriments. S'adapta al canvi, ja que els requeriments poden anar canviant després de cada entrega. D'aquesta manera, s'afegeix valor al producte.

Taula comparativa dels diferents enfocaments entre el model predictiu en cascada i el model iteratiu incremental.

Agile

Per tant, les principals avantatges que proporciona un model de gestió Agile són, primerament, la millora de la qualitat del producte, ja que fomenta la col·laboració dels membres de l'equip, que busquen i cooperen per obtenir l'excel·lència del producte. S'ha de tenir en compte que el fet de que es facin integracions i proves constants millora notablement el la qualitat i la fiabilitat del resultat final.

Seguidament, tenir client satisfet és un punt fonamental durant el procés de desenvolupament, i s'aconsegueix fent que cooperi activament, podent veure l'evolució del projecte mitjançant entregues ràpides de software amb valor.

Els treballadors estan més motivats, Agile es basa en equips de treball autogestionats, d'aquesta manera, cada membre de l'equip pot desenvolupar les seves qualitats creatives i innovadores per enriquir el projecte i alhora col·laborar amb diferents equips que es divideixen la feina i es reuneixen freqüentment.

Un altra avantatge important és la utilització de mètriques realistes; com que el treball es divideix en fases més petites, és més fàcil mesurar-lo a nivell de temps, cost i rendiment. Els resultats són més realistes.

Pel mateix motiu, les fases curtes o iteracions en les que es divideix el treball, hi ha més capacitat de predicció, es pot revisar i adaptar el producte a mida que avança el desenvolupament, d'aquesta manera, hi ha més control sobre la feina realitzada i permet millorar la capacitat de predicció i costos.

Finalment, la reducció de costos és vital, dins la gestió àgil s'elimina la possibilitat de fracàs absolut, perquè els errors es van identificant a mida que es va avançant, en comptes d'esperar a que estigui acabat i s'hagi realitzat tota la inversió.

3.2. Scrum

Scrum és una metodologia àgil de gestió de projectes que es basa en un marc de treball incremental per projectes de desenvolupament i en la col·laboració d'equips auto-gestionats multidisciplinaris.

3.2.2. Antecedents històrics

El concepte d'Scrum s'origina l'any 1986, amb la publicació d'un article de la Harvard Business Review, "The new new product development game"[7]. El terme Scrum s'extreu del Rugby (melée), quan l'equip forma una unitat al voltant de la pilota, lluitant per la seva possessió; els autors de l'article utilitzen aquesta metàfora per descriure el desenvolupament dels productes.

El contingut de l'article descriu com empreses com ara Honda, Canon i Fuji-Xerox produeixen nous productes utilitzant un nou enfoc de gestió, basat en equips integrals pel desenvolupament de productes, donant èmfasi al fet de que són equips auto-organitzats.

L'any 1993, Jeff Sutherland⁶ i el seu equip creen el procés d'Scrum per utilitzar-lo en l'àmbit del desenvolupament de software, barrejant conceptes de la programació orientada a objectes,

⁶ Jeff Sutherland: un dels inventors del procés de desenvolupament de software Scrum. Va col·laborar en la redacció del manifest Agile l'any 2001.

amb les de l'article de 1986, donant com a resultat un control de processos amb un desenvolupament iteratiu i incremental, i una millora de la productivitat en sistemes complexos i dinàmics.

Dos anys més tard, Ken Schwaber⁷ publica el primer informe sobre Scrum a OOPSLA⁸, formalitzant el procés per la indústria de desenvolupament de software.

A partir d'aquest moment Schwaber i Sutherland han publicat diverses especificacions per Scrum, incloent "Desenvolupament de Software Agile amb Scrum", "Gestió de Projectes Àgils amb Scrum" i "Guia d'Scrum".

Avui en dia, Scrum és utilitzat per milers d'empreses en milers de projectes, alguns exemples són els següents:

Sectors	Empreses
Telecomunicacions	BBC, Nokia, Motorola, Sony Ericsson.
Software i Hardware	Adobe, IBM, Intel, Microsoft.
Internet	Amazon, Google, Yahoo.
Banca	Bank of America, Key Bank, Merrill Lynch.
Sanitat	Philips Medical, Patientkeeper.
Videojocs	Blizzard, Ubisoft, Electronic Arts.

⁷ Ken Schwaber: desenvolupador de software, va treballar amb Jeff Sutherland en la creació de les primeres versions de Scrum com a procés de desenvolupament.

⁸ OOPSLA: Object-Oriented Programming, Systems, Languages & Applications, conferència de recerca anual.

3.2.3. *Actors, artefactes i cerimònies*

Per entendre millor el funcionament d'Scrum és necessari identificar i definir els diferents actors, artefactes i cerimònies que el caracteritzen.

Actors

Els principals actors que podem identificar en la metodologia Scrum són, primerament el product owner, el propietari del projecte, només n'hi ha un i és responsable del valor del treball. El seu principal objectiu és maximitzar el retorn de la inversió, és a dir, obtenir el màxim benefici de la seva inversió, identificant les característiques del projecte. Això significa traduir els objectius en històries d'usuari i col·locar-les en una llista prioritzada, decidint quins ítems han d'estar al capdamunt de la llista pel següent sprint, i reprioritzar i refinar constantment els elements de la llista.

Seguidament, trobem a l'equip, que es caracteritza per ser cross-functional (multi-funcional), és a dir, que inclou els experts necessaris per entregar el producte al final de cada sprint. Això vol dir que en un mateix equip trobem gent especialitzada en disseny, codificació, testing i desenvolupament, per exemple.

És un equip auto-organitzat, amb molta autonomia en la presa de decisions, que decideix en què es compromet i la millor manera d'aconseguir aquest compromís. Són equips de més o menys set persones.

Finalment, trobem l'Scrum Master, que és l'encarregat d'ajudar a l'equip i al product owner a comunicar-se i tenir èxit. Facilita el procés, donant suport a l'equip, escoltant les seves peticions i fent el seguiment dels sprint, liderant les principals cerimònies d'aquesta metodologia.

Si s'ha de parlar amb el product owner, l'Scrum Master és l'encarregat de fer-ho. És el pont entre l'equip i el product owner; tot i així, no és el líder de l'equip o un gestor de projectes, no li diu a la gent el que he de fer o com fer les diferents tasques.

Cerimònies

La principal cerimònia que caracteritza Scrum són els daily meetings. Són reunions de 5 a 15 minuts que es realitzen cada dia a una hora concreta, tots els membres de l'equip han de participar-hi i s'ha d'estar de peu, mai assegut.

Cada membre respon a les preguntes de l'Scrum Master, que és l'encarregat o encarregada de dirigir la reunió. Aquestes respostes fan referència a únicament tres coses:

1. Què s'ha fet des de l'últim Daily Meeting?
2. Què es farà de cara al pròxim Daily Meeting?
3. Hi ha hagut algun impediment o bloqueig?

Com que no hi ha un líder de l'equip, els daily meetings no són una eina per comunicar a un superior la feina que s'ha fet, simplement s'utilitza com a eina d'organització i informació per la resta de l'equip. Si hi ha algun problema que no respongui a aquestes tres preguntes, no es tracta durant el daily meeting.

Seguidament, una altra cerimònia característica és l'sprint review, que es realitza un cop s'ha acabat l'sprint. L'objectiu de l'sprint review és estudiar i adaptar la feina que s'ha fet durant l'sprint, generant un diàleg entre el product owner i l'equip. És el moment de veure si les eines i l'organització utilitzada en l'sprint anterior han sigut útils, i en cas de que no sigui així, modificar-ho i adaptar-ho. Si no s'ha acabat algun dels ítems que s'havien d'acabar durant l'sprint, es tornen a ficar al backlog.

Finalment, també quan es finalitza l'sprint, és realitza l'sprint retrospective, és una reunió entre l'equip i l'Scrum Master, on s'inspecciona i s'adapta el procés, es documenta el que funciona bé i el que es podria millorar; es fan propostes de millora i s'accepten si es necessari.

Artefactes

Els artefactes que es generen durant el procés són, en primera instància, el product backlog, que es caracteritza per ser una llista que existeix i implica tot el procés de vida del producte, és el mapa del producte, és una llista prioritzada de totes les característiques que ha de complir el projecte, prioritzades i constantment reordenades i perfilades pel product owner. Bàsicament és tot allò que l'equip podrà fer, ordenat per prioritat. Només existeix un product backlog.

Per tal de realitzar el seguiment del procés i poder visualitzar com avança, s'utilitzen els Burndown Charts, que són uns gràfics que es realitzen al final de cada sprint, on podem comptabilitzar l'esforç de l'equip. L'objectiu és arribar al sobrant d'esforç zero en la data de finalització de l'sprint.

Per generar aquest gràfic, en cada daily meeting es recullen dades sobre les hores que queden segons les estimades per cada integrant de l'equip.

Aquesta informació s'ha de formalitzar en una taula com la de la figura 3.

Ítem	Tasques de l'sprint	Voluntari	Esforç inicial estimat (hores)	Dia 1	Dia 2	Dia 3	Dia 4 (finalització sprint)
ITEM 1	Tasca1.1	Membre 1	8	4	3	0	0
	Tasca1.2	Membre 2	13	13	8	5	0
	Tasca1.3	Membre 3	5	5	5	2	0
ITEM 2	Tasca2.1	Membre 4	5	3	8	8	8
	Tasca2.2	Membre 5	3	3	3	3	3

Figura 3. Exemplificació d'una taula de seguiment d'hores durant un sprint.

I el gràfic resultant és similar a aquest:

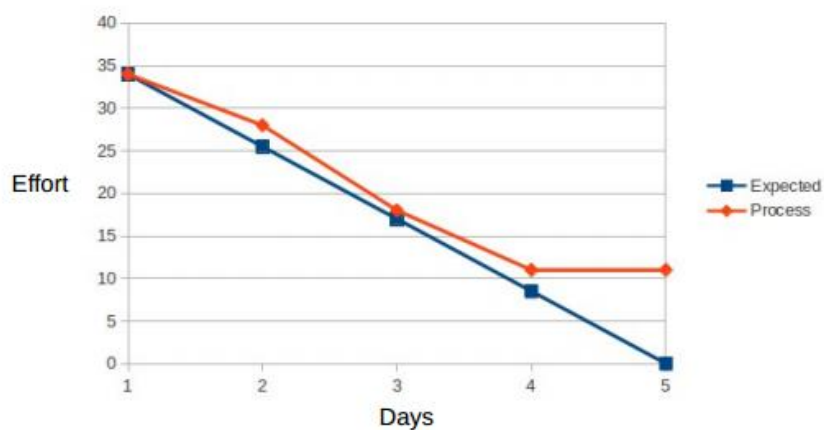


Figura 4. Burndown Chart.

A la figura 4 podem veure un Burndown Chart que no arriba a l'esforç sobrant zero i per tant ens indica que no s'han acabat totes les tasques planificades per l'sprint.

3.2.4. Esquema

Per acabar amb l'estudi teòric d'Scrum s'adjunta un esquema explicatiu del cicle de vida del projecte mitjançant Scrum, tenint en compte cerimònies, actors i artefactes.

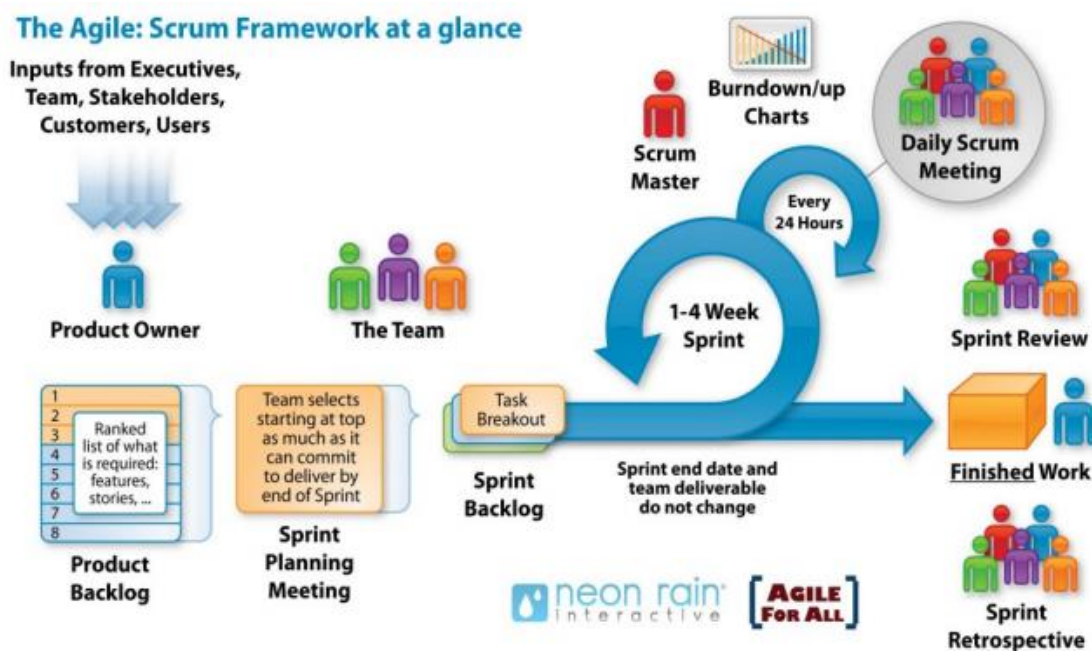


Figura 5. Esquematzació del cicle de vida d'un projecte que aplica Scrum com a metodologia de gestió de projectes.

A la figura 5 podem observar el cicle de vida d'un projecte aplicant Scrum. S'inicia amb el Product Owner i l'elaboració del Burndown Chart, seguidament la planificació de l'sprint i l'obtenció del sprint backlog. Un cop seleccionades les tasques per a realitzar durant l'sprint, d'una durada de entre 1 a 4 setmanes, es realitza diàriament un daily meeting per realitzar un seguiment del procés de producció. Un cop acabat l'sprint es realitza l'entrega d'una part del producte acabada i l'sprint review i l'sprint retrospective.

3.2.5. Discussió

Si ens centrem en l'aplicació d'Scrum en el desenvolupament de software, veiem que és un gran exemple de metodologia àgil, ja que compleix gran part dels seus principis, com ara la creació de software lliurable ràpidament, amb poca documentació inicial, equips multifuncionals amb la potestat de prendre les decisions necessàries, i per tant una manca de jerarquies, i finalment

la rapidesa de les iteracions en les que es divideix el desenvolupament del projecte, que permet al client fer modificacions i millores al llarg de tot el procés.

El treball es centra en un projecte que es divideix en sprints amb l'objectiu de millorar el rendiment, l'eficiència i l'adaptabilitat als canvis durant el procés de producció.

Aquesta organització de treball ens aporta la característica principal d'aquesta metodologia, que és "inspeccionar i adaptar-se"; ja que el procés de desenvolupament sempre implica aprenentatge, innovació i adaptació a imprevistos, Scrum aposta per repetir aquestes fases dins de cada sprint, inspeccionant el producte resultant i l'eficàcia de les pràctiques aplicades, adaptant, seguidament, aquestes pràctiques a les metes del producte. Aquesta filosofia s'aplica fins que s'acaba el procés de desenvolupament i es considera que el producte està acabat i és entregable.

El funcionament principal d'Scrum és al divisió del procés en sprints, i la realització d'un seguit de cerimònies durant el procés de desenvolupament, com ara els daily meetings, els sprint reviews i Burndown Charts, entre altres, que s'analitzaran en profunditat al punt 3.2.3.

Aquesta organització aporta una entrega de millor software més ràpid, aportant un enfoc creatiu al procés de desenvolupament, motivant als seus treballadors.

Scrum és aplicable per qualsevol nivell de complexitat i s'escala linealment al nombre de programadors que s'impliquen en el projecte.

Els sprints tenen una sèrie de característiques remarcables; primerament, han de ser curts, com a màxim poden durar un mes, i s'han de realitzar sense pausa. Tenen dates específiques, tant d'inici com de finalització, i no s'admeten variacions ni extensions. Dins de cada sprint es treballa amb un o varis requeriments del product owner, que s'obtenen d'una llista prioritzada d'elements anomenada backlog, per tant, l'objectiu principal de l'equip és entregar, al final de cada sprint, un software testeable i funcional basat en les peticions prioritzades del product owner.

Durant aquests sprints, el ítems seleccionats per desenvolupar no canvien, i cada dia es revisa l'evolució d'aquestes tasques; de la mateixa manera, al final de cada sprint es realitza una revisió de la feina feta, obtenint feedback dels progressos i els problemes.

3.3. Kanban

Kanban és una paraula japonesa que significa tarja visual, l'element clau d'aquest mètode de gestió; un taulell i varis post-it's o targetes seran els representants del flux de treball del projecte.

3.3.2. Antecedents històrics

L'any 1945, un cop acabada la segona guerra mundial, l'economia japonesa va quedar en un estat desastrós, pràcticament arruïnada i amb una tecnologia obsoleta per falta de diners i innovacions durant els anys de guerra. Tot i així, aquesta situació de crisi, com en molts casos, va ser el factor que va impulsar la busca de l'eficiència dels processos productius, iniciant noves tècniques, mètodes i sistemes de manufactura per evitar el malbaratament de matèries primeres.

Durant els anys 50, l'empresa Toyota, va trobar una manera de gestionar el flux de materials en una línia de producció, des d'aquest moment, es coneix Kanban com un sistema de producció altament efectiu i eficient, amb una reducció de costos considerable i una entrega d'artefactes excel·lent.

Durant un viatge a Estats Units, un grup d'enginyers japonesos es van fixar en el funcionament dels supermercats, interpretant els contenidors vuits de les seccions del supermercat com a ordres (s'ha d'omplir). Aquesta simple idea va despertar la idea de la tarja o etiqueta a mode d'instrucció, i posteriorment la idea del pull; no s'afegeixen nous aliments als contenidors fins que no estiguin vuits o a un mínim establert. Això es va convertir en: no s'afegeixen noves tasques de producció al taulell fins que no s'acabin les que s'estan fent (pull, don't push).

Des d'aquest moment, les tècnica es va desenvolupar molt ràpidament a Japó, més concretament a Toyota, que ja havia ideat un mètode de visualització del flux de treball.

A partir de 1958, va començar a funcionar correctament i la generalització d'aquesta idea va esdevenir en el sistema Kanban, més tard generalitzat a altres àmbits com el del desenvolupament de software, un àmbit amb molts punts en comú amb el de la manufactura en gran empreses.

3.1.1 *Actors, artefactes i cerimònies*

Especificant el mètode Kanban al sector que ens ocupa, és a dir, el de desenvolupament de software, podem identificar actors, artefactes i cerimònies.

Actors

Com en el cas d'Scrum, trobem el paper de product owner, només n'hi ha un i també és l'encarregat de definir les tasques del backlog, prioritzar-les i perfilar-les durant el procés de desenvolupament del projecte.

En el cas de Kanban, és completament necessari que les tasques èpiques del product backlog es divideixin en subtasques, que són les que es visualitzaran i viatjaran a través del taulell amb les diferents fases del projecte.

A part del product owner, trobem l'equip de desenvolupament, que és l'encarregat de crear les subtasques a partir de les històries èpiques d'usuari i desplaçar-les al llarg del taulell, sempre i quant compleixin les polítiques necessàries per passar a la següent fase.

També són els encarregats de redactar, juntament amb el product owner, aquestes polítiques d'acceptació, que són un llistat de factors que la tasca ha de complir abans de passar a la següent fase.

La principal característica dels equips en el model de Kanban és la integració de la filosofia de la mateixa metodologia; amb això ens referim a que, al ser un mètode menys estricte que Scrum, que té daily meetings i sprint amb dates d'inici i de finalització, Kanban no fixa límits de temps, per tant és necessari que l'equip sigui conscient de que, tot i no tenir una pressió a nivell temporal, s'ha de treballar al màxim rendiment sempre, de manera organitzada i prioritzant sempre l'entrega ràpida de software al client.

També són equips auto-gestionats i multidisciplinaris.

Cerimònies

La principal cerimònia dins de Kanban és el pas d'una tarja a una nova fase de producció.

Per realitzar aquest pas s'ha de complir que la tarja representi una història d'usuari no èpica i que compleixi les polítiques explícites definides a la part inferior del taulell.

Un cop s'ha passat una tarja a una altra fase, podem ficar-ne una de nova a la fase on estava inicialment, sempre respectant el WIP, que s'indica amb una xifra en cada columna, i així fins que la columna que conté les tasques del product backlog s'ha quedat buida i es considera que el treball està acabat i s'han complert totes les polítiques.

3.1.2 Esquema

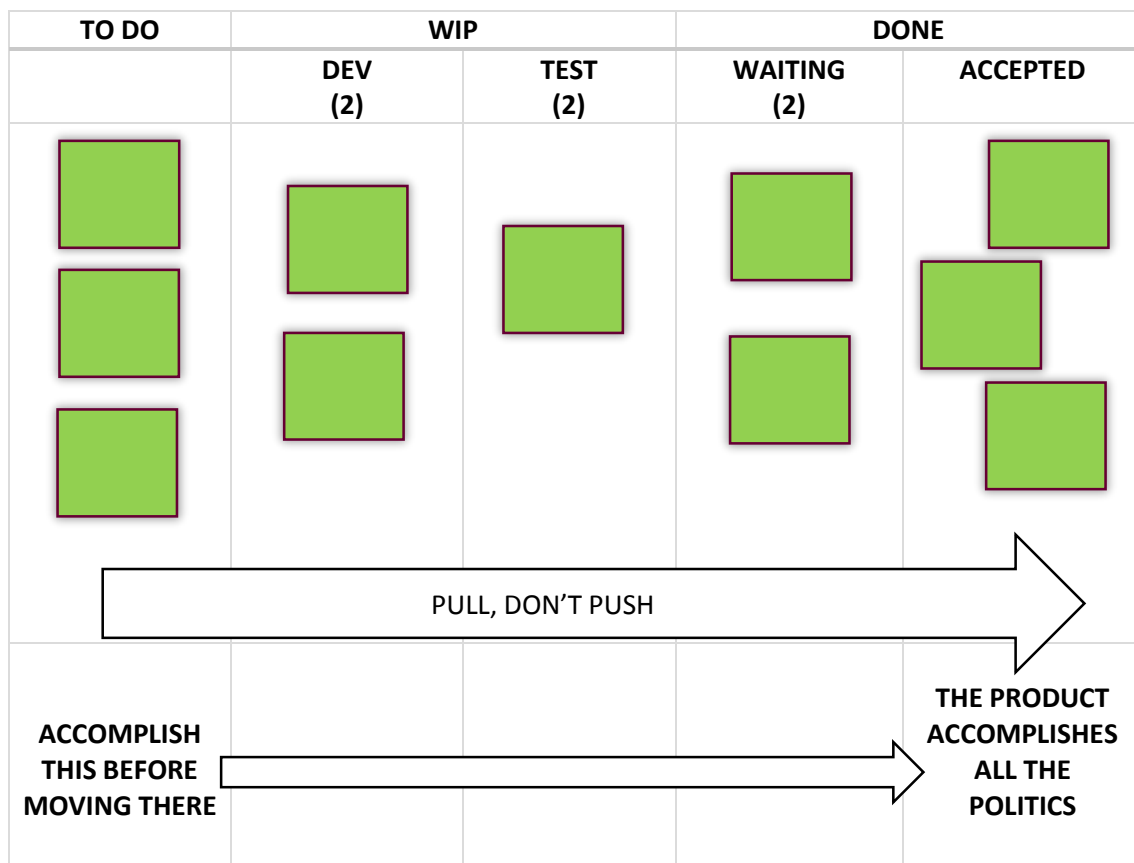


Figura 6. Esquematització del taulell Kanban.

A la figura 6 podem observar una esquematització del característic taulell que s'utilitza en Kanban per realitzar la visualització del flux de treball i l'evolució del projecte. Veiem

representades les característiques de la limitació del WIP mitjançant nombres a les columnes, el compliment de polítiques i la filosofia pull.

3.1.3 Discussió

Kanban divideix el procés de producció en fases ben delimitades, representades per les columnes del taulell. Dins de cada fase trobem tasques, representades mitjançant targes, que han de complir una sèrie de polítiques per poder passar a la següent fase, garantint d'aquesta manera, un producte de qualitat.

D'aquest sistema, inicialment aplicat a la indústria de l'automòbil, David J.Anderson⁹adapta la filosofia per aplicar-la al desenvolupament de software, amb diferents fases i equips de treball.

Els dos principals objectius amb els que es basa Kanban són, d'un costat, aconseguir un producte de qualitat, obligant a cada fase del projecte a finalitzar-se correctament abans de passar a la següent, i per altra banda, acabar amb el caos i els colls de d'ampolla durant les etapes de desenvolupament en les que prima la rapidesa per sobre de la qualitat del producte.

Kanban es fonamenta en quatre principis:

1. Comença amb el que estàs fent ara: fa referència a analitzar la feina que s'està fent actualment, i poder valorar si s'està fent bé o malament. Kanban no és un sistema que diu com fer la feina, simplement dona les eines necessàries per visualitzar-la i organitzar-la.
2. Accepta el canvi: si alguna de les coses que s'estan fent no funciona, s'ha de canviar, i si alguna cosa pot funcionar millor, es millora. Tots els membres de l'equip han d'estar disposats a aplicar canvis constants per millorar la seva feina, ja que tot canvi és una millora pel sistema. Kanban anima els canvis incrementals i evolutius.
3. Accepta el procés en curs, els rols i les responsabilitats de cadascú: cada membre de l'equip ha de ser conscient de què ha de fer i de quines són les seves funcions. Perquè Kanban funcioni, és necessari que quedi clar que no es tracta de que tothom faci de tot, sinó de que cadascú sigui conscient del que ha de fer en el moment adequat.
4. Lideratge en tots els nivells: l'equip ha d'estar ben gestionat i tenir iniciativa pròpia. No s'han de crear sistemes piramidals amb burocràcia entre equips, sinó que cada subgrup

⁹ David J.Anderson: principal formulador del mètode Kanban.

i cada membre ha de tenir clar el seu punt de lideratge i realitzar la feina que li pertoca correctament.

Per aplicar correctament Kanban, s'han de seguir unes pautes o pràctiques principals.

Aquestes pràctiques es poden dividir en 10 passos fonamentals, que es poden fer servir com una guia per aplicar Kanban correctament.

Primerament, s'ha de poder **visualitzar el flux de treball**. Aquesta visualització és la base per veure com avança el projecte, si no s'entén aquest flux, és molt complicat fer canvis.

Per representar aquest flux de treball s'utilitzen taules dividides en columnes que representen els diferents estats del procés, mentre que les targetes representen cada tasca, i es desplacen per les columnes. Com a norma bàsica, Kanban considera que només podem gestionar el treball que podem visualitzar.

El segon punt es basa en la limitació del nombre de tasques que es fan alhora, és a dir, **limitar el Work In Progress (WIP)**, que és el nombre de tasques en procés en el sistema. Fer moltes coses alhora i no fer-ne bé del tot cap no serveix de res, val més acabar una cosa correctament abans de començar-ne una altra.

Per tal de tenir una mesura de les tasques que es van realitzant es defineix l'anomenat Cycle time, que és el WIP / les tasques realitzades per unitat de temps.

$$\text{CYCLE TIME} = \text{WIP} / \text{THROUGHPUT}$$

Reduir el cycle time incrementa el flux i minimitza la quantitat de treball i el risc.

En la limitació del WIP també és important tenir en compte la mida de les tasques que es realitzaran i l'aplicació de la metodologia pull ("Pull, don't push"), que fa referència a no afegir noves tasques al taulell fins que s'acabi una, evitant els colls d'ampolla i assegurant una bona gestió del flux de treball.

Dirigir i gestionar el flux de treball és vital per una bona aplicació d'aquest model, i això no vol dir només visualitzar el flux de treball, sinó també mesurar-lo i controlar el seu funcionament.

Per assegurar la integritat i bon funcionament de les tasques que es van desenvolupament, s'han de deixar clares les **polítiques d'acceptació** en cada fase, de manera que si una tasca no les compleix, no pot passar a la següent fins que no ho faci. D'aquesta manera, amb unes polítiques explícites i ben definides ens estalviem gran part dels errors que cometríem sense un model de

gestió explícit. S'ha d'aconseguir que aquestes polítiques siguin visibles constantment, i s'han de modificar quan és necessari, intentar millorar el sistema implica qualitat en el resultat, i especificar i variar les polítiques va directament relacionat amb una millora del sistema.

El quart punt fa referència a trobar un **bon ritme d'entrega de software funcional**, s'ha de determinar la durada dels cicles per tal de mantenir sempre el ritme d'entrega i el rendiment de l'equip.

Seguidament, **la mesura del flux de treball**, s'han d'escollir mètriques correctes per a la mesura del progrés, es poden utilitzar, per exemple, Cumulative Flow Diagrams, que són una substitució del Burndown Charts utilitzats en Scrum. A part, dins de la mesura del flux de treball, també s'ha de tenir en compte la visualització constant dels elements que bloquegen el flux, fer-los visibles en el taulell de Kanban i buscar una solució, ja sigui dividint-los en tasques més petites o fent que més membres de l'equip treballin en aquesta tasca.

Seguidament, **la prioritització**, s'aconsegueix ordenant els ítems del backlog i escollint els que estan a dalt de tot de la llista, és a dir, els més prioritaris.

Troblem factor importants de prioritització, com ara:

- El risc i la incertesa. S'han de detectar les tasques que comporten un alt risc o que representen un gran impacte en el procés de desenvolupament.
- Detectar les necessitats bàsiques del projecte, com ara la seva infraestructura, organització i procés d'evolució.
- Balancejar històries d'usuari. S'han de barrejar històries d'usuari amb un valor alt de negoci amb aquelles que no en tenen per tal de mantenir constant el flux de valor. Si es prioritzen totes les tasques amb valor de negoci i no es tenen les altres en compte, en un inici el valor de negoci del software entregat serà molt alt, però un cop s'hagin realitzat aquestes tasques, el valor caurà en picat.
- Dependències. Si existeixen dependències entre les històries d'usuari s'han de prioritzar de manera que es realitzin de forma ordenada sense generar elements bloquejants.

El sisè punt fa referència a la definició de tipus de servei, és a dir, **definir el tipus de treball** que es realitza durant el procés de desenvolupament. Trobem, com a tasques a realitzar, les històries d'usuari, bugs, tasques de suport, instal·lacions varies etc. Cada tipus de tasca representa un cost concret i un tractament específic, per exemple, si hi ha un bug en el procés de desenvolupament que bloqueja tot el flux de treball, se li ha d'atorgar una prioritat màxima i el

cost de solucionar-lo serà elevat. Per tant, és útil diferenciar les diferents tasques i ordenar-les segons la seva prioritat, cost o tractament.

El setè punt posa l'accent en la **gestió del flux de treball**, és a dir, entendre el sistema i prendre decisions apropiades de millora. Per tal de gestionar correctament el flux de treball és útil fer-nos preguntes com les següents:

- El WIP està ben limitat?
- Es poden fer les històries d'usuari més petites?
- Hi ha alguna manera de detectar les tasques que acaben sent molt més costoses un cop s'estan realitzant?
- Es pot trobar una manera d'augmentar la flexibilitat del sistema per evitar elements bloquejant i per tant colls d'ampolla?
- S'està intentant optimitzar la totalitat del sistema i no els elements individuals?

La clau per gestionar correctament el flux de treball és pensar sempre en el flux de treball final, el que fa referència al producte final, i intentar no centrar-se en fer més ràpid un pas individual, és a dir, pensar sempre de forma global centrant-se en el producte final.

El novè pas es basa en la **millora continua del sistema**; crear un sistema capaç d'adaptar-se als canvis i utilitzar-los com una avantatge competitiva i una millora del producte final, ser capaç de veure oportunitats de millora tant del producte com del procés de desenvolupament.

Finalment, la **millora en equip**; un dels pilars de Kanban és la millora constant, i aquesta ha de ser acordada per l'equip, aportant l'experiència de tots els membres.

3.2 Lean

La metodologia Lean enfocada completament en el client, busca identificar el valor de negoci, reduir malbarataments, crear un flux basat en mesures, posposar la presa de decisions i entregar software amb rapidesa i perfecció.

3.2.1 Antecedents històrics

L'any 1867, neix Skichi Toyoda, fill d'un fuster, viu a un poble de pagesos, on les dones treballen amb telers. Duran la seva joventut, utilitza tots els seus coneixements per modernitzar el teler que utilitza la seva mare, i l'any 1891 patentava el seu primer teler automàtic i es muda a Tokio, per començar un nou negoci de telers. Passen els anys, i al 1896 firma un contracte per comercialitzar els telers, molt més econòmics que a Alemanya i França.

Passades les guerres amb Xina (1894) i Japó (1904), l'any 1929, després de tornar d'Estats Units i interessar-se per la novetat de l'automòbil, Sikichi Toyoda ven les seves patents de telars i encarrega al seu fill, Kiichiro Toyoda, a invertir en la indústria automotriu, sembrant les bases de la corporació Toyota. L'any 1931, Sikichi mor i Kiichiro, tres anys més tard, funda la Divisió Automotriu Toyota Automatic Loom Works.

L'any 1937, Kiichiro produeix el primer prototip d'automòbil i acaba sent conegut com el "Rei dels inventors japonesos". Un any més tard crea la primera planta de producció a gran escala, implementant el concepte de producció JIT (Just In Time): produir només allò que es necessita, en el moment que es necessita i en les quantitats necessàries, reduint els nivell d'estoc i aconseguint estalvis considerables a nivell de costos.

A partir d'aquest moment, la producció i expansió de la marca Toyota creix a nivell mundial, i a principis dels anys 90, James Womack escriu "La màquina que va canviar el món", inspirada en les pràctiques iniciades per Kiichiro, i que ha acabat sent una Bíblia per les empreses interessades en millorar els processos de producció.

L'evolució d'aquesta filosofia de producció s'ha expandit en molts àmbits laborals, no només en cadenes de producció, sinó també en el món del desenvolupament de software, per acabar sent la filosofia que coneixem ara.

3.1.1 *Actors, artefactes i cerimònies*

Donat que la metodologia Lean aplicada al desenvolupament de software no és molt clara, és més una filosofia d'empresa que una metodologia en si, la detecció d'actors, artefactes i cerimònies s'ha realitzant partint de les bases de les metodologies àgils, pensant en una bona adaptació de la filosofia de reducció de malbaratament i prioritització del valor de negoci.

Per aplicar Lean en aquest projecte, s'ha utilitzar una segona metodologia per a poder seguir el flux de treball, Kanban.

Actors

Identifiquem com a actors dins de la metodologia Lean, primerament a l'equip, encarregat de seleccionar les tasques del backlog i desenvolupar-les en cicles iteratius i incrementals, i el product owner, encarregat de definir el product backlog i prioritzar les tasques segons valor de negoci.

L'equip treballa aplicant constantment, les solucions proposades per reduir el malbaratament (Seven Wastes of Lean) durant la producció, i prioritzen sempre les tasques que aporten valor de negoci.

Artefactes

A nivell d'artefactes, existeix el Lean canvas, que és un taulell on es defineix la idea principal de l'empresa i del projecte. El taulell, dividit en varis segments ens aporta informació sobre el problema que ha de resoldre el producte, la solució proposada, les mètriques claus que es faran servir per mesurar el procés i els guanys, la proposició única de valor, l'avantatge competitiva en relació a altres empreses, els canals de distribució, els clients als que anirà dirigit i que podran oferir subvencionant, l'estructura de costos i el flux d'ingressos.

En el cas concret d'aquest projecte, també torbaríem el taulell de Kanban per visualitzar el flux de treball i el backlog prioritzat.

Cerimònies

Finalment, Lean no especifica cap tipus de cerimònia, però es consideren bàsiques les reunions entre l'equip i el product owner per definir les històries d'usuari, prioritzar-les segons valor de negoci i redefinir el backlog.

3.1.2 Esquema

La metodologia Lean aplicada al desenvolupament de software no té una definició clara, per tant, es presentaran dos figures diferents.



Figura 7. Esquematització del cicle de vida d'un projecte en un equip que aplica Lean com a model de gestió.

A la figura número 7 podem veure les fases definides per Eric Ries¹⁰ que s'apliquen quan s'utilitza Lean. En aquest cas podem veure la concordança entre el procés de vida d'un projecte Lean amb el cicle de vida iteratiu i incremental, fonamental en les metodologies àgils.

¹⁰ Eric Ries: autor i emprenedor, pioner del moviment Lean Startup, estratègia que dirigeix les companyies startup per tal de treballar amb els seus recursos de la manera més òptima possible.

2 PROBLEMA	4 SOLUCIÓ	3 PROPOSICIÓ ÚNICA DE VALOR	9 AVANTATJA COMPETITIVA	1 SEGMENTS DE CLIENTS
	8 MÈTRIQUES CLAU		5 CANALS DE VENTA	
7 ESTRUCTURA DE COSTS			6 FLUX D'INGRESSOS	

Figura 8. Esquematització del Lean Canvas.

A la figura número 8 podem veure el Lean Canvas, on es defineix la idea general del projecte.

3.1.3 Discussió

Els tres pilars fonamentals d'aquesta metodologia són l'enfoc i orientació completa cap al client, l'aportació de valor al client, eliminant el malbaratament, és a dir, tot allò que no aportí valor de negoci, i la flexibilitat en el procés de desenvolupament, amb la finalitat de respondre de manera àgil a la demanda del mercat i les seves variacions.

L'origen de Lean com a metodologia de gestió de projectes s'inicia amb Lean Manufacturing, i és un conjunt de principis que tenen com a objectiu aconseguir qualitat, velocitat i complir les expectatives del client. Les seves principals premisses són eliminar residus, assegurar al qualitat del producte, crear coneixement dins de l'equip, endarrerir el compromís, posposant la presa de decisions, entregar ràpid, apostar per l'autonomia individual i optimitzar el sistema.

Bàsicament, busca eliminar tot allò que no agrega valor de negocia al projecte, centrant-se únicament en allò necessari en l'actualitat.

Quan parlem d'eliminar tot allò que no agrega valor de negoci, ens referim a eliminar reunions innecessàries, tasques secundàries, documentació sense valor i maneres ineficients de fer la feina.

A nivell d'equip, Lean s'esforça en aconseguir que l'equip funcioni com una unitat, sabent que tota aportació individual s'optimitza per la resta de l'equip; tot i així, defensa l'autonomia de cada persona dins de l'equip.

Com s'ha comentat en un inici, un dels objectius de Lean és crear coneixement dins de l'equip, per tant, l'aprenentatge és una part important, i es fomenta de manera continuada, i és per aquest motiu que es retarda la presa de decisions, perquè mentre no s'està prenent la decisió, es considera que l'equip, i cada individu que el forma, va aprenent, i per tant, la solució que s'ofereix quan es pren la decisió és la que té més qualitat i valor.

La perfecció i la qualitat dels elements produïts és essencial en la cultura Lean, i tots els processos s'han de centralitzar en obtenir aquesta qualitat i perfecció; d'aquesta manera es guanya avantatge competitiu.

Com s'ha comentat, un altre dels pilars essencials en la metodologia Lean és l'eliminació de tot allò que no aporta valor de negoci. Per eliminar aquest malbaratament s'han definit els anomenats "Seven Wastes of Lean", i són els següents:

1. Feina parcialment acabada: fa referència al treball que un cop entregat no compleix la definició de Done, i per tant no es vàlid per a ser testejat. Pot incloure codi no optimitzat, testejat, integrat, ben documentat o que simplement no es pot desplegar.
Alguns dels motius pels quals pot passar això són, per exemple, la prioritització d'una història d'usuari sense tenir tota la informació sobre el que ha de fer, la complexitat tècnica no s'ha analitzat degudament, dependències amb altres històries d'usuari etc.
2. Afegir més característiques de les necessàries: significa produir més del que es demana. I pot passar per una mala comprensió de la visió del producte, una mala prioritització de les característiques del producte, una mala especificació d'aquestes etc.
3. "Relearning": és a dir, no utilitzar el coneixement disponible dins de l'equip, tornar a inventar la roda quan el company del costat ja té els plànols. Això pot ser degut a una mala comunicació entre l'equip, la manca d'un procés d'intercanvi de coneixements, falta de documentació o equips distribuïts.
4. "Hands-off": fa referència al moment en que se li passa la feina a una altra persona un cop s'ha completat una part; aquest canvi ha de ser ràpid i intuïtiu, per evitar el Waste nº 3. El motiu pel qual el passar la feina a un altre membre de l'equip no funcioni bé pot ser, per exemple, la manca de visibilitat de la informació i de l'estat de desenvolupament durant el procés, mala comunicació entre l'equip o la falta de documentació un cop acabada una part del projecte.

5. Retards: són qualsevol cosa que impliqui trigar més en entregar el projecte, o dedicar molt temps a tasques que no tenen valor de negoci. Poden ser deguts a una manca de membres especialitzats dins de l'equip, masses coses al WIP, dependències externes o no tenir clar el concepte de valor de negoci.
6. Canvi de tasques: els membres es desplacen d'una tasca a una altra sense completar-les correctament. Això pot ser degut a interrupcions durant el desenvolupament d'una tasca, mala coordinació entre el product owner i els membres de l'equip o un equip que treballa en més d'un projecte alhora.
7. Defectes: funcionalitats errònies que generen un output erroni, ja pot ser per mala comprensió de les històries d'usuari, les històries no compleixen el principi INVEST[17], falta de polítiques explícites, de criteri d'acceptació o un testeig massa tard.

4. POSADA EN PRÀCTICA. DESENVOLUPAMENT.

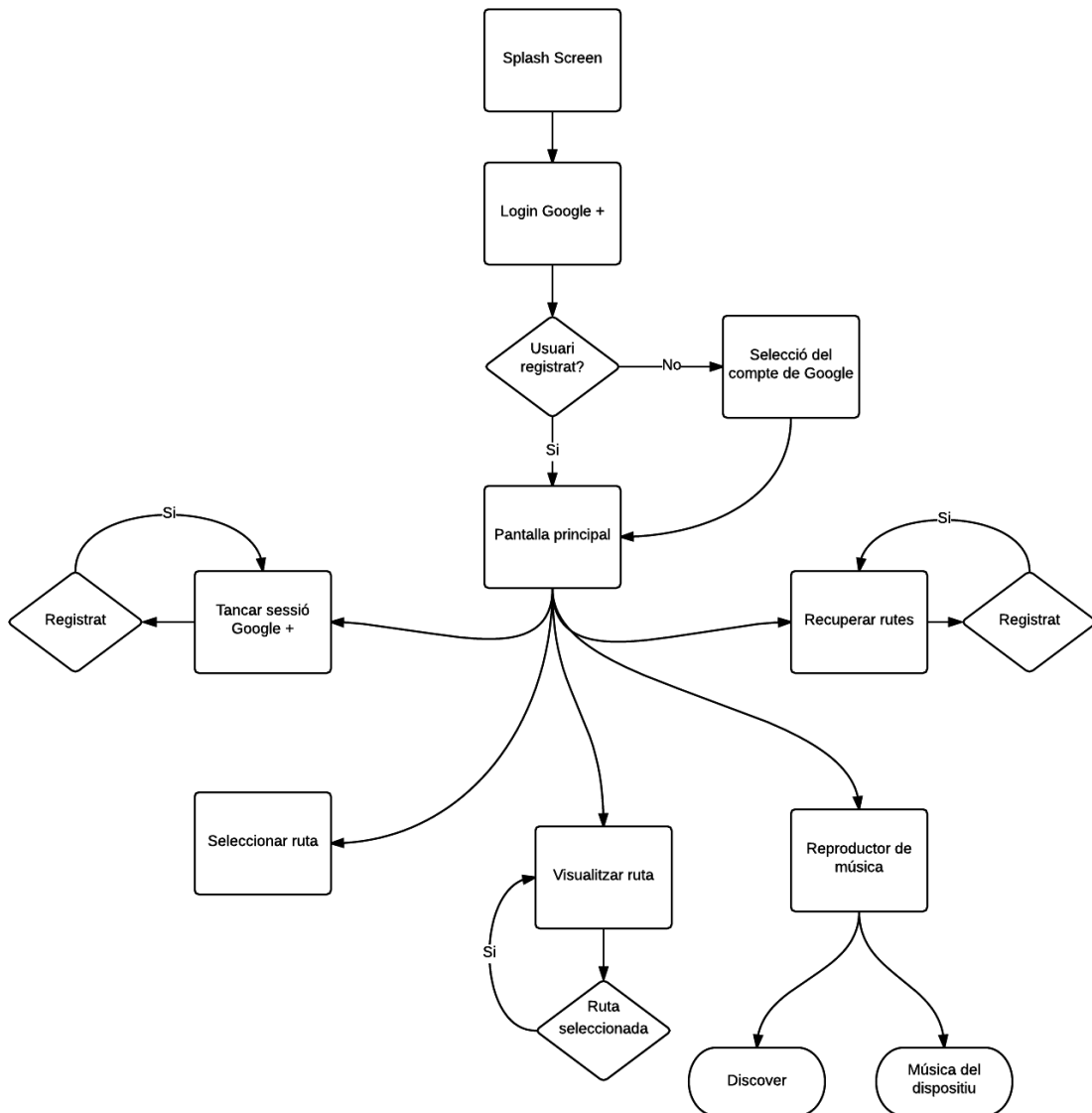
En aquest apartat es presentarà la posada en pràctica de les diferents metodologies presentades anteriorment per al desenvolupament de l'aplicació Android.

Abans de començar amb l'anàlisi de l'aplicació teòrica és necessari recordar la idea i funcionalitat de l'aplicació.

La idea principal a desenvolupar és la d'una aplicació Android que acompanyi a l'usuari en una ruta a peu. Ofereix un servei per visualitzar la ruta sobre un mapa de Google Maps, podent escollir el punt inicial i final de la ruta; a part, fa un seguiment de la distància, temps i velocitat de l'usuari mentre realitza la ruta i ofereix indicacions mitjançant un sintetitzador de veu. Un cop seleccionada la ruta, permet previsualitzar-la mitjançant imatges de Street View i finalment, ofereix un reproductor de música, de manera que mentre l'usuari realitza la ruta té dues opcions, o fer servir el reproductor per escoltar les cançons del seu dispositiu o veure quines cançons estan escoltant altres usuaris al seu voltant.

A part, l'aplicació ofereix la possibilitat de guardar i recuperar les rutes desitjades i registrar-se mitjançant Google Plus.

A nivell esquemàtic, la navegabilitat de l'aplicació és la següent:



- Quan l'usuari accedeix a l'aplicació veu l'Splash Screen amb el logo principal de l'aplicació.
- Seguidament, apareix la pantalla de registre amb Google Plus, si l'usuari ja ha escollit un compte anteriorment, i per tant ja està registrat, l'aplicació passarà directament a la pantalla principal. En cas contrari, l'usuari seleccionarà un compte de Google per registrar-se.
- Un cop a la pantalla principal, l'usuari pot accedir a les diferents subaplicacions, mitjançant el menú o seleccionant les icones.
 - a. Selecció d'una ruta.

- b. Previsualització d'una ruta amb imatges de Street View si hi ha alguna ruta seleccionada (ja sigui recuperada o acabada de seleccionar).
- c. Reproductor de música, on podrà escollir escoltar la música del dispositiu o del demés usuaris que estiguin fent servir l'aplicació.
- d. Recuperar rutes de base de dades.
- e. Tancar la sessió amb Google Plus.

Un cop presentada la navegabilitat, podem extreure els requeriments principals que ha de complir l'aplicació:

1. Seleccionar una ruta dinàmicament i visualitzar-la sobre el mapa.
2. Oferir un seguiment durant la ruta, donant indicacions en punts crítics i mesurant la distància, el temps i la velocitat de l'usuari.
3. Poder previsualitzar la ruta amb imatges de Street View.
4. Oferir un reproductor de música per escoltar la música del dispositiu o d'altres usuaris que estiguin utilitzant l'aplicació.
5. Guardar i recuperar rutes.
6. Registrar-se mitjançant Google Plus.

A nivell pràctic, l'aplicació desenvolupada, anomenada ROAD, es basa en tres subaplicacions; la primera ens permet escollir una ruta mitjançant un mapa de Google Maps, escollint el punt d'origen i el punt de finalització de la ruta. Dins d'aquesta mateixa activitat, queda a disposició de l'usuari l'opció de guardar la ruta a base de dades i també un botó per iniciar la ruta. Un cop s'ha finalitzat la ruta, l'usuari prem el botó per indicar que ha acabat, i l'aplicació mostra el temps que ha estat caminant, la distància recorreguda i la velocitat amb que ha recorregut la totalitat o una part de la ruta. A part, també disposa d'indicacions orals, de manera que l'aplicació "parla" quan l'usuari s'acosta a punts on es necessària una senyalització per a seguir la ruta.

La segona subaplicació de ROAD és l'opció de visualitzar la ruta que es va a realitzar. Aquest apartat s'ha realitzat descarregant imatges de Google Street View i mostrant-les mitjançant un timer, de manera que dona la sensació de veure un vídeo de la ruta.

La tercera subaplicació permet a l'usuari accedir a la música del seu dispositiu i fer servir l'aplicació com un reproductor de música normal, però a part, també dona l'opció de veure que està escoltant la gent que rodeja l'usuari i mostrar-li una llista ordenada segons proximitat, de

manera que en primera posició apareixerà la cançó de la persona més propera. Evidentment aquest cas s'ha simulat, ja que seria necessari que molt usuaris fessin servir aquesta aplicació per tal de realitzar-ho realment. Tot i així, l'única modificació necessària seria allotjar la base de dades creada localment a un servidor al qual accedirien tots els usuaris.

Finalment, també es dona l'opció de veure les rutes guardades, accedir al menú d'ajuda i tancar o iniciar la sessió amb Google Plus.

Les tecnologies utilitzades per realitzar l'aplicació són, a nivell d'entorn de desenvolupament Android Studio, un IDE (Integrated Development Environment) basat en IntelliJ de JetBrains, dissenyat específicament pel desenvolupament d'Android

Per a repositori de dades s'ha utilitzat Bitbucket, sistema de control de versions Mercurial i Git.

A continuació es presentarà cada metodologia adaptada a la situació concreta del projecte, els punts febles i els punts forts que s'han observat, les conclusions personals i fragments de codi importants referents a cada part de l'aplicació desenvolupada.

Per cada metodologia de gestió s'han dedicat tres setmanes de desenvolupament.

4.1 Scrum

4.1.1 Disseny de la FASE 1

La part de l'aplicació escollida per a ser gestionada amb Scrum és la selecció d'una ruta mitjançant un mapa de Google Maps i l'inici de l'aplicació, incloent inici i finalització de la sessió de Google Plus i interfície d'usuari bàsica.

A nivell de codi, aquesta primera part de l'aplicació implica la creació de tres activitats Android diferents, una per mostrar el mapa, seleccionar i guardar les rutes, una altra per fer el login i el logout de Google i finalment l'activity principal que serà l'encarregada de mostrar el menú principal de l'aplicació i redirigir a l'usuari a les diferents subaplicacions.

Una activity d'Android és una classe dins del nostre projecte Android que fa extens de la classe Activity, i es presenta a l'usuari com una finestra que mostra una interfície d'usuari. Una activity té diferents estats, pot estar executant-se, en pausa o parada. Està en execució quan es visible o s'interacciona amb ella, està pausada quan és visible però una altra finestra ocupa tota la

pantalla o s'executa en un pla superior i està parada quan no és visible. En tots aquests casos, la classe manté la seva informació.

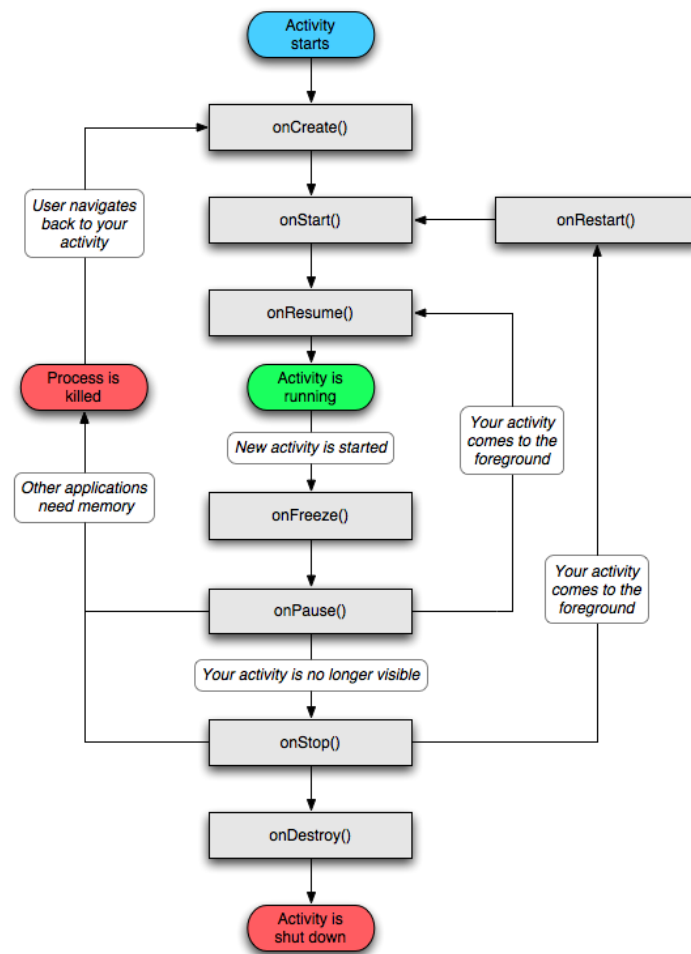


Figura 9. Gràfic que representa els diferents estats de vida i d'execució d'una activitat. [19]

A la figura 9 podem veure una esquematització dels diferents estats de vida d'una activitat d'Android, en una primera instància veiem les crides a les funcions per iniciar-la, després les funcions que s'executen quan l'activitat ja no és visible, i finalment les funcions per quan es surt de l'aplicació, passa en un segon pla, o l'usuari tanca l'aplicació.

Per realitzar aquest primer apartat de l'aplicació, seguint la metodologia Scrum, es va desenvolupar un backlog prioritzat per saber quines eren les funcions que havia de complir, almenys en un primer termini, aquesta primera part.

Per tal de poder definir les subtasques i prioritzar-les dins del backlog, es va començar per definir una història d'usuari èpica:

- Aplicació que permeti escollir a l'usuari una o varies rutes via Google Maps i que l'aplicació el vagi notificant durant la ruta.

Partint d'aquesta història èpica es van definir les següents històries d'usuari:

- Com a usuari vull un inici de sessió mitjançant Google Plus al iniciar l'aplicació
 - a. Vull que només la primera vegada hagi d'escollir el compte de Google a utilitzar.
- Com a usuari vull que les rutes que creï amb el meu compte de Google es quedin guardades i hi pugui accedir més tard.
- Com a usuari vull poder ficar un nom a les rutes que guardi.
- Com a usuari vull tenir a la meva disposició un log out del meu compte de Google.
- Com a usuari vull escollir el punt d'inici i punt de finalització de la ruta
- Com a usuari vull visualitzar la ruta al mapa.
- Com a usuari vull una interfície simple i intuïtiva.

Un cop definides les històries d'usuari de la primera part de l'aplicació, es van redactar les subtasques de cadascuna. Es van realitzar tres sprints, un sprint per setmana.

El primer sprint, del 19/03 al 26/03 incloïa les següents subtasques:

1. Relatiu a la interfície bàsica de l'aplicació [Simple i intuïtiva]
 - a. Selecció d'icones
 - b. Aplicació d'una estètica uniforme
 - c. Realització de l'Splash Screen [apareix el logo principal de l'aplicació]
2. Creació d'una base de dades de tipus SQLite
 - a. Realitzar la primera connexió amb base de dades
3. Login i logout mitjançant compte de Google
 - a. Guardar l'usuari a base de dades
4. Creació d'un repositori Bitbucket
5. Habilitar Android Studio per a testejar en dispositius [incloent primer contacte amb IDE]

Tal i com indica Scrum, durant l'sprint es van realitzar daily meetings cada dia, tot i que la situació del projecte no permetia fer daily meetings en equip, es van realitzar a nivell individual. La repriorització del backlog es realitzava al finalitzar cada sprint a la reunió amb el tutor del projecte.

Durant els daily meetings efectuats al llarg de la setmana es van recollir les següents dades:

DAILY MEETING 20/03/2015			
Què s'ha fet des d'ahir?	Què es farà demà?	Impediments	Temps dedicat
Creació del repositori a Bitbucket i primer commit.	Test de l'aplicació en un dispositiu.	Llibreries de Google desactualitzades	5 hores
Primer contacte amb Android Studio.	Test de login amb Google.	Actualització de complements de l'SDK	
Creació d'una aplicació base.	Creació i inserció d'un usuari a BBDD.	Problemes de sincronització entre bitbucket i Android Studio	

DAILY MEETING 21/03/2015			
Què s'ha fet des d'ahir?	Què es farà demà?	Impediments	Temps dedicat
Detecció de dispositius per a fer tests.	Test de l'aplicació en un dispositiu.	Drivers de Windows 8.1 incompatibles per a connectar dispositius	3 hores
	Test de login amb Google.	Manca de permisos	
	Creació i inserció d'un usuari a BBDD.		

DAILY MEETING 23/03/2015			
Què s'ha fet des d'ahir?	Què es farà demà?	Impediments	Temps dedicat
Executar aplicació sobre dispositiu.	Relació entre navigation drawer i activity principal.	Cap	3 hores
Splash Screen.	Login via Google.		
Creació d'ícones.	Creació i inserció d'un usuari a BBDD.		
Esquematzació de la interfície bàsica.			
Creació de navigation drawer [interfície bàsica].			

DAILY MEETING 24/03/2015			
Què s'ha fet des d'ahir?	Què es farà demà?	Impediments	Temps dedicat
Modificacions de la interfície [logo, backgrounds, comportament].	Test de login amb Google.	Codi deprecated en la versió de l'api de Google.	6 hores
Creació Activity de login.		Obtenció SH-1.	
Obtenció SH-1 per Google Api.		Actualitzacions vàries.	
Connexió amb Google Plus.			

DAILY MEETING 25/03/2015			
Què s'ha fet des d'ahir?	Què es farà demà?	Impediments	Temps dedicat
Login mitjançant Google.	Revisió del correcte funcionament del login i planificació de proper sprint.	Cap	2 hores

DAILY MEETING 26/03/2015			
Què s'ha fet des d'ahir?	Què es farà demà?	Impediments	Temps dedicat
Modificacions de la interfície.	Log out de Google Plus.	Problemes amb la modificació del drawable layout.	5 hores
Creació de base de dades.	Inserció de dades a base de dades.		
Comprovació de connectivitat amb Google Plus.	Començar a treballar amb localització dins de l'activity.		
Segon sprint planning.			

Un cop acabat el primer sprint, tal i com indica Scrum, es va realitzar la taula amb el recompte d'hores relatiu a cada daily meeting i la creació del Burndown Chart.

A continuació es mostren les tasques de l'sprint enumerades per a una millor comprensió de la taula del recompte d'hores i del Burndown Chart.

Tasca 1: Interfície gràfica [Icones, estètica, splash screen].

Tasca 2: Creació de base de dades.

Tasca 3: Login mitjançant compte de Google.

Tasca 4: Logout del compte de Google.

Tasca 5: Creació del repositori bitbucket.

Tasca 6: Habilitació del repositori de bitbucket.

Tasca	Dia 1	Dia 2	Dia 3	Dia 4	Dia 5	Dia 6	Dia 7
1	10	10	8	6	3	0	0
2	2	2	2	2	2	1	0
3	6	6	6	6	3	0	0
4	1	1	1	1	1	1	1
5	3	0	0	0	0	0	0
6	5	4	1	0	0	0	0

Figura 10. Sprint 1 - Aquesta taula inclou les estimacions de temps i la dedicació per a cada subtasca.

La figura 10 fa referència al recompte d'hores i les estimacions de temps realitzades per desenvolupar les tasques seleccionades en el primer sprint.

Com es pot observar, la tasca número quatre, referent al logout de Google no es va realitzar durant aquest sprint.

Aquesta evolució i l'esforç sobrant es poden veure representats al Burndown Chart que es mostra a continuació:

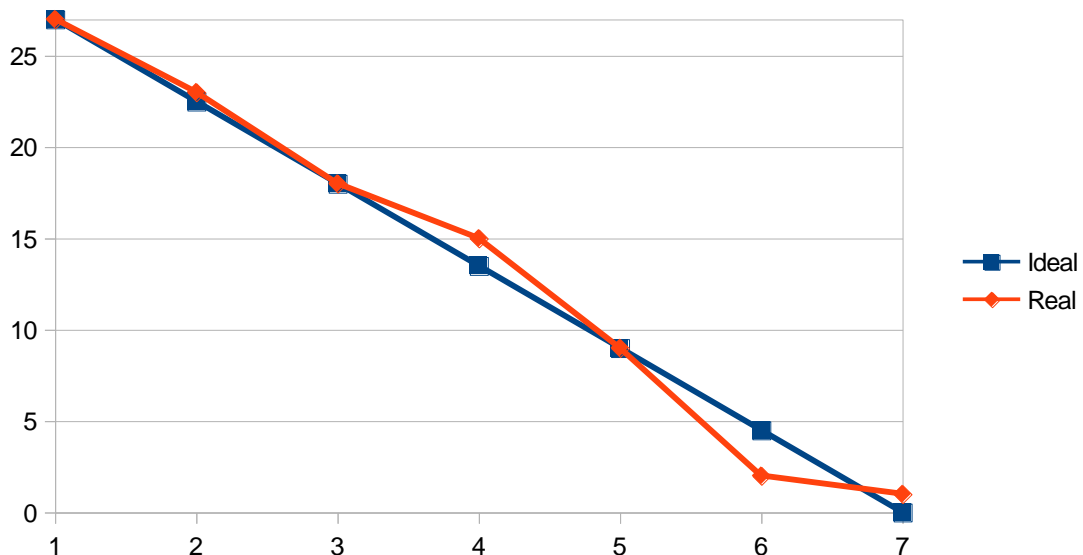


Figura 11. Sprint 1 - Representació gràfica mitjançant un gràfic de tipus Burndown de l'esforç realitzant durant un sprint de set dies.

La figura número 11 ens mostra l'esforça realitzat durant un sprint de set dies amb un sobrant d'esforç d'una hora, que fa referència a una tasca no realitzada.

Podem observar que queda un sobrant d'esforç d'una hora que fa referència al log out de Google Plus. També podem observar que el rendiment va baixar lleument el dia quatre, en el que es van dedicar menys hores de les previstes i per tant, menys tasques. Tot i així també podem veure que del dia 5 al dia 6 es van realitzar més tasques de les esperades, dedicant més hores de les estimades idealment. En conclusió, el rendiment és prou constant, proper al rendiment ideal.

Un cop acabat el primer sprint, es van definir les tasques per al segon sprint, que comprèn des del dia 27/03 al 02/04. Les tasques són les següents:

1. Iniciar api de Google Maps i fer-la funcional.
2. Fixar inici i final de ruta amb Google Maps [no dinàmic, per primer contacte].
3. Mostrar ruta al mapa [ruta a peu].
4. Seleccionar inici i final dinàmicament.
5. Mostrar localització actual.

Com es pot veure, no s'han inclòs els elements no acabats correctament de l'sprint anterior; tal i com indica Scrum, un cop acabat un sprint, totes les tasques no finalitzades tornen al backlog, i aquest es reprioritza abans d'iniciar el següent sprint.

Durant la repriorització es va considerar que les tasques no finalitzades tenien menys importància que les escollides en aquest sprint.

Els daily meetings d'aquest segon sprint són els següents:

DAILY MEETING 28/03/2015			
Què s'ha fet des d'ahir?	Què es farà demà?	Impediments	Temps dedicat
Creació activity de Google Maps i correcte funcionament.	Escollir ruta amb inici i final.	Cap	1 hora

DAILY MEETING 29/03/2015			
Què s'ha fet des d'ahir?	Què es farà demà?	Impediments	Temps dedicat
Mostrar mapa dins l'activity.	Millorar aspecte de la ruta i afegir markers.	Cap	4 hores
Fixar inici i final de ruta estàtic .	Seleccionar inici i final dinàmicament.		
Creació d'una tasca asíncrona per crear la ruta.			
Mostrar la ruta al mapa.			

DAILY MEETING 30/03/2015			
Què s'ha fet des d'ahir?	Què es farà demà?	Impediments	Temps dedicat
No s'ha pogut avançar.	No s'ha pogut avançar.	No s'ha pogut avançar.	0 hores

DAILY MEETING 31/03/2015			
Què s'ha fet des d'ahir?	Què es farà demà?	Impediments	Temps dedicat
No s'ha pogut avançar.	No s'ha pogut avançar.	No s'ha pogut avançar.	0 hores

DAILY MEETING 01/04/2015			
Què s'ha fet des d'ahir?	Què es farà demà?	Impediments	Temps dedicat
Mostrar markers.	Planificació del nou sprint.	Cap	3 hores
Mostrar localització actual.			
Mostrar indicacions.			
Mostrar distància restant.			
Selecció dels punts d'inici i final dinàmicament.			

Un cop finalitzat l'sprint es va realitzar la taula amb el recompte d'hores i el Burndown Chart.

Tasques:

1. Iniciar api de Google Maps i fer-la funcional.
2. Fixar inici i final de ruta.
3. Mostrar ruta a peu al mapa.
4. Seleccionar inici i final dinàmicament.
5. Mostrar indicacions.
6. Mostrar localització actual.

Tasca	Dia 1	Dia 2	Dia 3	Dia 4	Dia 5	Dia 6	Dia 7
1	4	3	0	0	0	0	0
2	1	1	0	0	0	0	0
3	6	6	0	0	0	0	0
4	3	3	3	3	3	0	0
5	3	3	3	3	3	0	0
6	3	3	3	3	3	0	0

Figura 12. Sprint 2 - finalitzades totes les tasques previstes per aquest sprint.

La figura 12 fa referència al recompte d'hores i les estimacions de temps realitzades per desenvolupar les tasques seleccionades en el segon sprint.

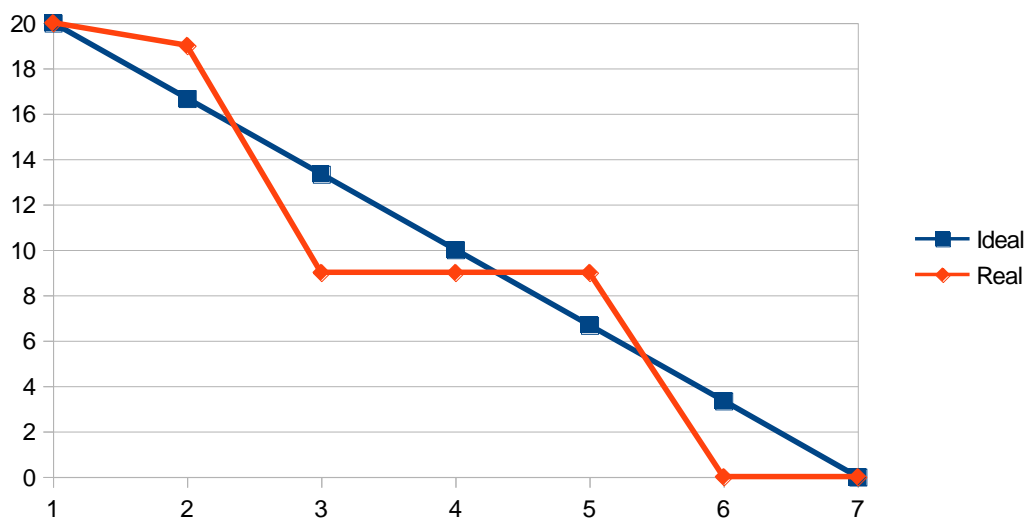


Figura 13. Sprint 2 - Burndown Chart amb esforç sobrant 0. S'han completat totes les tasques.

La figura número 13 ens mostra l'esforça realitzat durant un sprint de set dies amb un sobrant de zero hores que significa que s'han realitzat totes les tasques.

Com es pot observar, la representació del gràfic concorda perfectament amb els daily meetings. Durant dos dies (els dies 30 i 31 de març) no es realitza cap avanç i per tant l'esforç queda representat amb una línia constant. La resta de dies es realitza més feina de la prevista, compensant, d'aquesta manera, els dos dies d'aturada.

El rendiment en aquest sprint ha sigut completament irregular, havent de fer tota la feina en pocs dies en comptes de dividir-la durant la setmana.

Les estimacions de temps, clarament, no han sigut acurtades en aquest sprint; es va suposar que es trigaria molt més en fer la feina. En la majoria dels casos, s'han realitzat varies tasques en el temps estimat per fer-ne només una.

Si que és veritat que tota metodologia àgil recomana fer estimacions pessimistes, però en aquest cas no han sigut prou realistes.

De cara al tercer i últim sprint, comprès entre el dia 02/03 i el 09/03, es va definir el següent llistat de tasques a realitzar:

1. Posicionament [tracking de la posició actual] i zoom inicial.
 - a. Amb tracking ens referim a fer un seguiment de la posició de l'usuari mentre es desplaça.
2. Informació durant la ruta.
 - a. Sintetitzador de veu.
 - i. Temps.
 - ii. Distància.
 - iii. Indicacions.

Els daily meetings són els següents:

DAILY MEETING 02/04/2015			
Què s'ha fet des d'ahir?	Què es farà demà?	Impediments	Temps dedicat
No s'ha pogut avançar.	No s'ha pogut avançar.	No s'ha pogut avançar.	0 hores

DAILY MEETING 03/04/2015			
Què s'ha fet des d'ahir?	Què es farà demà?	Impediments	Temps dedicat
No s'ha pogut avançar.	No s'ha pogut avançar.	No s'ha pogut avançar.	0 hores

DAILY MEETING 05/04/2015			
Què s'ha fet des d'ahir?	Què es farà demà?	Impediments	Temps dedicat
No s'ha pogut avançar.	No s'ha pogut avançar.	No s'ha pogut avançar.	0 hores

DAILY MEETING 06/04/2015			
Què s'ha fet des d'ahir?	Què es farà demà?	Impediments	Temps dedicat
Zoom inicial.	Tracking de la posició.	Cap	3 hores

DAILY MEETING 07/04/2015			
Què s'ha fet des d'ahir?	Què es farà demà?	Impediments	Temps dedicat
Tracking segons current position.	Que text to speech (sintetitzador de veu d'Android) indiqui distància recorreguda i temps.	No es detecta last position del dispositiu.	4 hores
Sintetitzador de veu [direccions].		S'ha hagut d'implementar un timer per emular el comportament del canvi de posició que ofereix la api de Google.	
Timer per refrescar posició.			

DAILY MEETING 08/04/2015			
Què s'ha fet des d'ahir?	Què es farà demà?	Impediments	Temps dedicat
Mostrar distància total de la ruta.	Tancament d'sprint.	Cap	3 hores
Eliminar algunes indicacions de veu [molest].			
Càlcul del temps caminat.			
Càlcul de distància recorreguda.			
Sintetitzador de veu: distància i temps.			

La taula de seguiment d'hores i el Burndown Chart són els següents:

Tasques:

1. Zoom inicial.
2. Tracking.
3. Temps caminat (amb sintetitzador de veu).
4. Distància restant.
5. Indicacions.
6. Distància total.

Tasca	Dia 1	Dia 2	Dia 3	Dia 4	Dia 5	Dia 6	Dia 7
1	3	3	3	3	0	0	0
2	7	7	7	7	7	0	0
3	2	2	2	2	2	2	0
4	2	2	2	2	2	2	0
5	2	2	2	2	2	0	0
6	2	2	2	2	2	2	0

Figura 14. Sprint 3 - finalitzades totes les tasques previstes per aquest sprint.

La figura 10 fa referència al recompte d'hores i les estimacions de temps realitzades per desenvolupar les tasques seleccionades en el tercer sprint.

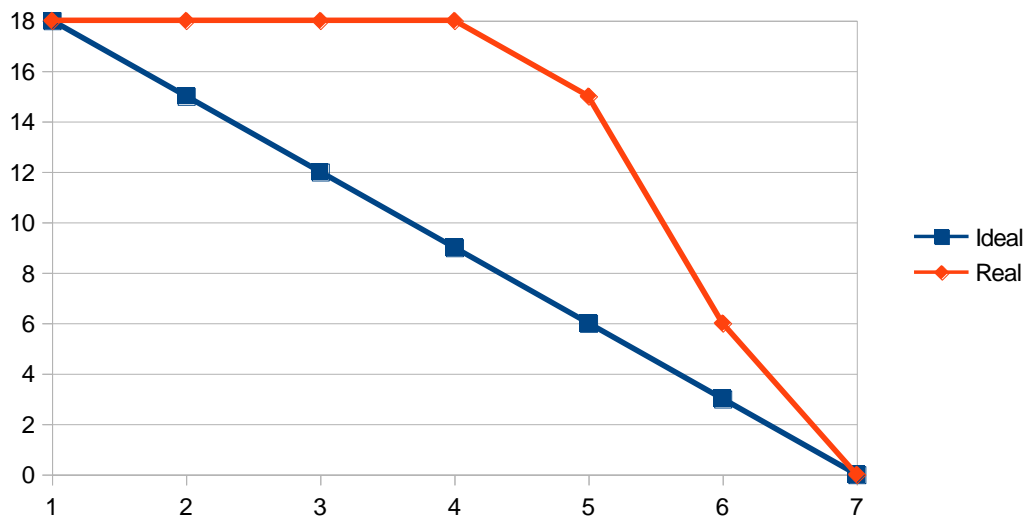


Figura 15. Sprint 3 - Burndown Chart amb esforç sobrant 0. S'han completat totes les tasques.

La figura número 15 ens mostra l'esforça realitzat durant un sprint de set dies amb un sobrant de zero hores que significa que s'han realitzat totes les tasques.

Com es pot observar, el desenvolupament no es va poder iniciar fins el quart dia de l'sprint, tot i així, l'esforç sobrant és zero, ja que es va dedicar el temps suficient els altres dies.

4.1.2 Característiques aplicades

Les principals característiques que s'han pogut aplicar correctament amb l'aplicació d'Scrum en aquesta part de l'aplicació han sigut, primerament, els daily meetings; tot i no realitzar-se en equip i de peu, tal i com detalla Scrum, s'han pogut realitzar daily meetings cada dia i s'ha pogut efectuar un recompte d'hores i un seguiment del treball realitzat.

També s'ha pogut aplicar correctament el tancament d 'sprint, juntament amb l 'sprint review/retrospective.

Recordem que el tancament d 'sprint es centra en estudiar i adaptar la feina que s'ha fet durant l 'sprint, en aquest cas, durant la setmana. És un diàleg entre el product owner i l'equip, i es el moment de retornar al backlog les tasques no realitzades i reprioritzar-les.

L 'sprint review és una reunió entre l'equip i l'Scrum Master, on s'inspecciona el procés i es fan propostes de millora.

Per al desenvolupament d'aquest projecte, l'encarregat de fer de product owner és el tutor del treball, i per tant amb el que es discuteix la repriorització del backlog i també amb el que s'inspecciona el producte testeable que s'entrega al final de cada sprint i es discuteixen les possibles millores de cara al proper sprint.

També es van redactar correctament les històries d'usuari, partint d'una èpica, es va anar desglossant en subtasques desenvolupables que posteriorment es van afegir al backlog del producte, que com assenyala Scrum, va ser el mapa de ruta de tota la primera part de l'aplicació.

A part, també s'ha pogut veure la modificació de característiques amb les variacions de la interfície, en comptes de seguir al peu de lletra la idea original, es van anar realitzant modificacions a mida que s'anava avançant, d'aquesta manera, es pot veure la flexibilitat que ofereix Scrum davant d'aquestes situacions.

Finalment, a nivell d'artefactes, com s'ha pogut constatar, es van poder realitzar els Burndown Charts partint de les taules de recompte d'hores, on s'hi inclouen les estimacions de temps i el temps real dedicat.

En conclusió, tot i que al ser un treball individual, gran part de l'essència d'Scrum, que fa referència als equips auto-gestionats multidisciplinaris, s'ha perdut, la resta de característiques incloent cerimònies, artefactes i organització (característica referent a la divisió del cicle de vida en tres sprints, un per setmana), s'han pogut realitzar correctament amb resultats satisfactoris.

4.1.3 Conclusions personals

Un cop acabada l'experiència amb Scrum es van detectar varis punts positius i negatius basats en l'aplicació que s'ha fet en aquest projecte.

A nivell de punts positius, primerament, s'ha pogut observar que Scrum és una base sòlida per organitzar projectes de qualsevol tipus, ja siguin grans o petits i que impliquin una o varies persones. Ofereix una bona base ja que permet tenir una llista prioritzada de les tasques a realitzar, i permet escollir el nombre de tasques que es volen realitzar durant l'sprint, factor que atorga molta llibertat i permet una organització pròpia en base al temps disponible i les capacitats de cada persona. A nivell personal, opino que el fet de que una persona tingui la

potestat d'escollir el volum de feina que vol realitzar el converteix en una persona que treballa amb un millor rendiment, ja que no es satura o s'avorreix degut a una gran o mínima quantitat de feina que li ha atorgat algú que no té un coneixement de les seves capacitats; per tant també fa que la persona es mostri més animada i motivada de cara als projectes.

Seguidament, un altre punt molt positiu és la prioritització de les tasques, ja que dona una idea bàsica de quines han de ser les característiques mínimes del projecte un cop acabat l'sprint; permet no malgastar temps en tasques secundàries, i fa que la part del projecte que s'entrega un cop acabat l'sprint compleixi unes característiques bàsiques pel client, assegurant qualitat i un producte testejat.

Referent als daily meetings, són una manera excel·lent per realitzar un seguiment de la feina realitzada, tot i que a vegades, si les subtasques no estan prou ben definitives, en el sentit de que són massa genèriques, els daily meetings perden pes, ja que es fa referència a la mateixa tasca durant varis daily meetings sense veure el progrés real.

A nivell d'equip, és vital la realització dels daily meetings, és una cerimònia molt útil de cara al Scrum Master, ja que li permet fer un seguiment de la feina realitzada i comunicar-ho al product owner, incloent les estimacions de temps de cada membre de l'equip.

A nivell individual, òbviament, la realització dels daily meetings perd importància a nivell pràctic, ja que ja es té consciència de la feina feta el dia anterior; tot i així, sense la realització dels daily meetings seria impossible, inclús a nivell individual, realitzar la taula amb les estimacions de temps i per tant, els Burndown Charts.

En relació als Burndown Charts, em semblen un artefacte molt interessant, ja que es poden extreure conclusions molt interessants si es planteja, en una empresa, fer un estudi a gran escala; es pot veure quins dies són més productius i quins ho són menys segons cada membre de l'equip o de l'equip en general.

A part, també és una bona manera de visualitzar el progrés del projecte i per detectar les tasques que no s'han realitzat correctament, també serveix per veure si les estimacions de temps han sigut realistes, i així equilibrar el nombre de tasques de cara a futurs sprints.

Moltes vegades es tendeix a seleccionar masses tasques, i no es tenen en compte els impediments que poden aparèixer durant el procés de desenvolupament; els Burndown Chart permeten preveure aquestes situacions i seleccionar més o menys tasques del backlog.

A nivell de punts negatius, s'ha pogut veure que Scrum, tot i ser una bona i sòlida metodologia àgil per a la gestió de projectes, té algunes mancances, com ara incentius durant el desenvolupament.

No hi ha una visualització real i constant del flux de treball, de manera que una mala subdivisió de tasques, pot donar la sensació de que no s'avança en el procés.

Personalment opino que seria molt interessant que els membres de l'equip disposessin d'altres artefactes per visualitzar diàriament el flux de treball, i em sembla vital una bona subdivisió de tasques.

Crec que la visualització del flux de treball hauria de ser una base primordial en qualsevol metodologia de gestió de projectes, ja que també és una manera d'incentivar al treballador, ja que veu com evoluciona la seva feina o com s'estanca, i d'aquesta manera rendir més quan es necessari.

Finalment, a nivell d'aplicar Scrum per a projectes individuals, tot i que es perdi la característica principal de l'equip, continua sent molt interessant. Permet organitzar-se correctament el temps que s'hi dedica i fixar límits de temps als que s'ha de respondre. A part, poder visualitzar, mitjançant els Burndown Charts, el rendiment d'un mateix és una eina molt interessant per estudiar l'esforç que s'hi dedica i comprovar si les estimacions de temps fetes són correctes o no.

En conclusió, crec que Scrum aporta una base sòlida per a la realització de projectes, amb artefactes molt interessants que ofereixen un anàlisi, si es vol, a gran escala. Tot i així, opino que si es valorés més la visualització constant del flux de treball, la motivació i rendiment millorarien.

4.1.4 Resultats

Un cop finalitzada la primera fase de l'aplicació utilitzant Scrum com a metodologia àgil de gestió de projectes, el resultat és el següent:



Figura 16. Splash Screen de l'aplicació



Figura 17. Inici de sessió amb Google Plus

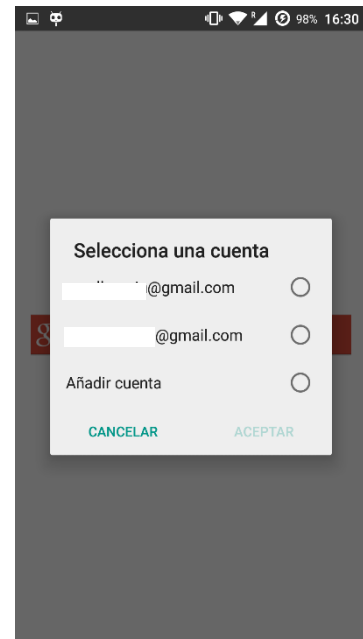


Figura 18. Selecció del compte de Google

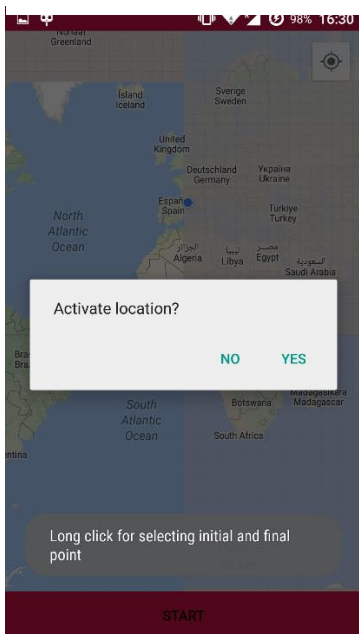


Figura 19. Localització en el mapa de Google.

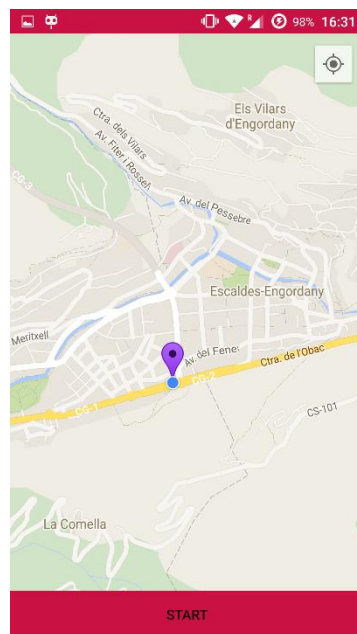


Figura 20. Zoom inicial amb senyalització de localització

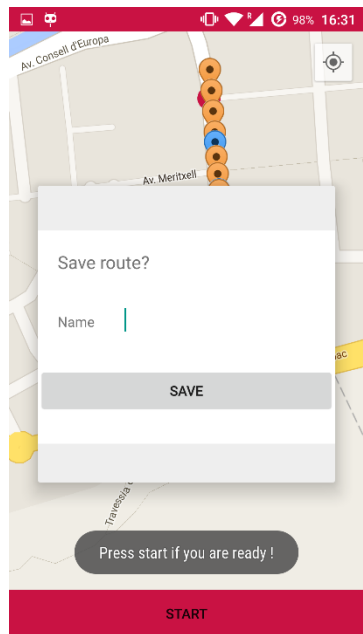


Figura 21. Selecció dinàmica de la ruta i opció per guardar-la

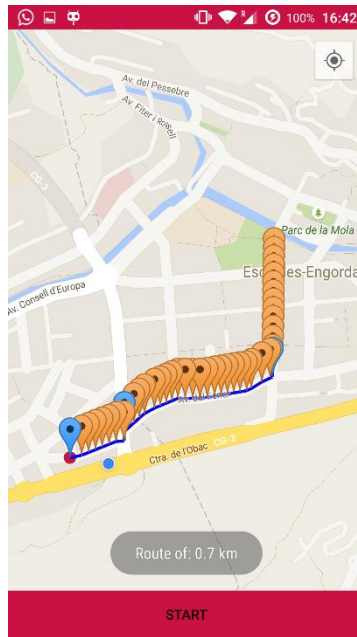


Figura 22. Visualització de la ruta a peu

La figura número 16 ens mostra la pantalla de presentació de l'aplicació, les figures 17 i 18 ens mostren l'inici de sessió mitjançant Google Plus, i finalment, les figures de la 19 a la 22 ens mostren l'activitat de selecció de ruta.

Per tal de realitzar aquesta part es van definir les següents classes que es poden visualitzar a l'annex número 1 d'aquesta memòria:

- Vista
 - PathGoogleMapActivity.java
 - Mostra la finestra amb el mapa de Google, s'encarrega de totes les inicialitzacions necessàries pel mapa i de la localització i el zoom inicial de la càmera.
 - Es crea una tasca asíncrona després de seleccionar el punt d'inici i final de la ruta que genera la ruta sense saturar el thread principal de l'aplicació.
 - Per generar la ruta es crea una URL on afegim el punt inicial i el punt final de la ruta i es seleccionen diverses opcions que ens ofereix la api de Google.
 - En un inici la url es genera en format String, creant un http request, i posteriorment es transforma en un objecte de tipus Json per realitzar el dibuixat a sobre del mapa.
 - Un cop obtinguda i "parsejada" la ruta, es dibuixa a sobre del mapa utilitzant un objecte de tipus Polyline que ofereix la api de Google.
 - Mostra els missatges corresponents (distància de la ruta, dialog per guardar la ruta i missatge de finalització de la ruta).

- Envia la informació corresponent a MensajeRuta.java mitjançant el controlador per configurar el sintetitzador de veu.
 - Controla el tracking de la posició mitjançant un timer que va comprovant i actualitzant la localització de l'usuari. Quan l'usuari s'acosta a un punt que necessita una indicació, s'encarrega de comunicar-li.
 - SplashScreen.java
 - Mostra el logo principal de l'aplicació i dona pas a l'activitat de login.
 - MainActivity.java
 - Conté un menú de tipus navigation drawer (lliscant des del lateral) i unes icones que permeten accedir a les diferents subtasques.
 - Inicia els Intents necessaris en cada cas.
 - LoginActivity.java
 - Inicialitza la connexió amb Google Plus mitjançant la api.
 - Mostra el botó de login o de log out segons cada cas.
 - Si l'usuari ja està loguejat mostra el botó de log out i al revés.
 - Tracta els casos de connexió fallida i suspena.
 - Si la connexió falla, torna a intentar-ho.
 - Quan s'inicia l'aplicació realitza el login sol si l'usuari ja ha seleccionat el seu compte un cop. Si és un usuari nou, li demana el compte de Google amb el que vol utilitzar l'aplicació.
 - Gestiona el tancament de sessió.
 - Quan l'usuari tanca la sessió, al tornar a iniciar l'aplicació o tornar a fer login, demana un altre cop el compte de Google.
 - S'han creat tots els xml corresponents a les activies declarades.
- Model
 - PathJSONParser.java
 - Crea l'HTTP Request.
 - Retorna l'url en format JSON.
 - MensajeRuta.java
 - Encarregat d'inicialitzar i gestionar l'objecte Text to Speech.
 - Fa els càlculs necessaris de la distància recorreguda per l'usuari i del temps que porta caminant.
- Controlador
 - Controlador.java

4.2 Kanban

4.2.1 Disseny de la FASE 2

Seguint amb el desenvolupament, la part escollida per aplicar Kanban ha sigut la subaplicació per visualitzar rutes descarregant les imatges de Google Street View i la inserció i recuperació d'aquestes a base de dades i la visualització de la ruta.

Per tal de realitzar la totalitat de les tasques correctament, es va crear una història d'usuari èpica, després es van definir tres històries d'usuari també èpiques que definien per parts els desenvolupament de la història èpica principal, i finalment, aquestes tres tasques èpiques es van dividir en subtasques desenvolupables.

La història èpica principal és:

- Com a usuari vull que l'aplicació em permeti visualitzar un vídeo de la ruta i recuperar les rutes que he guardat.

Un cop definida aquesta història èpica referent al funcionament genèric d'aquesta subaplicació, es van definir les següents "subhistòries èpiques d'usuari":

1. Com a programador, per simular el vídeo, vull descarregar fotos d'Street View amb la direcció i orientació correcta.
2. Com a usuari vull que, tot i que siguin fotos, estiguin composades de manera que doni la sensació de que és un vídeo.
3. Com a usuari vull guardar i recuperar les rutes que he creat.

La durada de l'aplicació de Kanban, igual que Scrum, va ser de tres setmanes, per tant, es va assignar una història d'usuari per setmana.

El backlog per a la primera setmana de desenvolupament, referent a obtenir les fotografies de Street View és el següent:

1. Buscar informació sobre Street View i integració amb desenvolupament Android.
 - a. Com indica Kanban, s'ha de tenir en compte la formació de l'equip i la dedicació d'hores a l'aprenentatge constant per obtenir resultats més sòlids i satisfactoris amb un millor rendiment.
2. Realitzar connexió amb Street View passant informació necessària a nivell estàtic (punt estàtic, direcció estàtica).

3. Obtenir punts equidistants de la ruta per descarregar les fotografies.
4. Esbrinar com calcula la orientació Street View i aplicar-ho.
5. Obtenir informació de tots els punts de la ruta.
6. Descarregar i guardar array de fotografies procedents d'Street View.

Per tal de realitzar totes aquestes tasques aplicant Kanban, és va comprar una pissarra i es va crear el taulell de Kanban per on van passant les tasques. Aquest taulell disposa de tres columnes principals:

- TO DO: és on trobem les tasques del backlog prioritzades. Són totes les tasques que s'han de realitzar al llarg de la setmana.
- WIP (Work in Progress): referent a les tasques que s'estan realitzant. Aquesta columna està dividida en dues sub-columnes, Developing i Testing. La columna de developing fa referència a les tasques que s'estan desenvolupant, i la de testing és on trobem les tasques que ja s'han desenvolupant i de les quals s'està fent el testeig.
- DONE: conté les tasques acabades. Està dividida en dues sub-columnes, waiting, on trobem les tasques que han passat el testeig però necessiten l'aprovació del product owner, i done, on hi ha les tasques que ja tenen l'aprovació del product owner i que per tant, estan finalitzades i són entregables.

Una altra part fonamental per aplicar correctament la metodologia àgil Kanban és la definició de polítiques explícites; són les normes o característiques que ha de complir una tasca per passar d'una columna a una altra.

La política definida per passar de TO DO a WIP és: Les subtasques estan prioritzades i no són èpiques.

D'aquesta manera, ens assegurem de que les tasques que passem a desenvolupar estan ben definides i són prioritàries. Només amb la definició d'aquesta política ja estem obtenint un millor rendiment, ja que no es perd temps en tasques no prioritàries, i obtenim una millor comprensió de la feina que s'ha de realitzar, ja que la tasca no és èpica i descriu exactament quina feina s'ha de fer.

La següent política, per passar de DEVELOPING a TESTING és que el desenvolupament es considera acabat, de manera que a priori, sense tenir en compte el testeig, el desenvolupament realitzar compleix les característiques de la tasca que estem realitzant.

Per passar de TESTING a WAITING és necessari que la tasca funcioni correctament i realitzi les funcions necessàries un cop superat el test.

Finalment, per passar de WAITING a DONE, és necessària l'aprovació del product owner, que considera que la tasca realitza tot el que s'espera d'ella.

Una altra característica important al aplicar Kanban, és la limitació del nombre de tasques que poden haver-hi en cada columna. En el cas d'aquest projecte, es va limitar el WIP a tres tasques, en concret, només podia haver-hi una tasca en desenvolupament i dos a la columna de testing. Aquesta elecció es va prendre tenint en compte que es desenvolupava l'aplicació a nivell individual, i per tant, com a molt podia es desenvolupar una tasca i provar-ne dues simultàniament.

També es va limitar el nombre de tasques que podien estar a waiting, ja que per complir la política i passar a done era necessària una reunió amb el tutor del projecte, i no es podia saturar excessivament aquesta columna.

El taulell amb el que s'ha treballat és el següent:

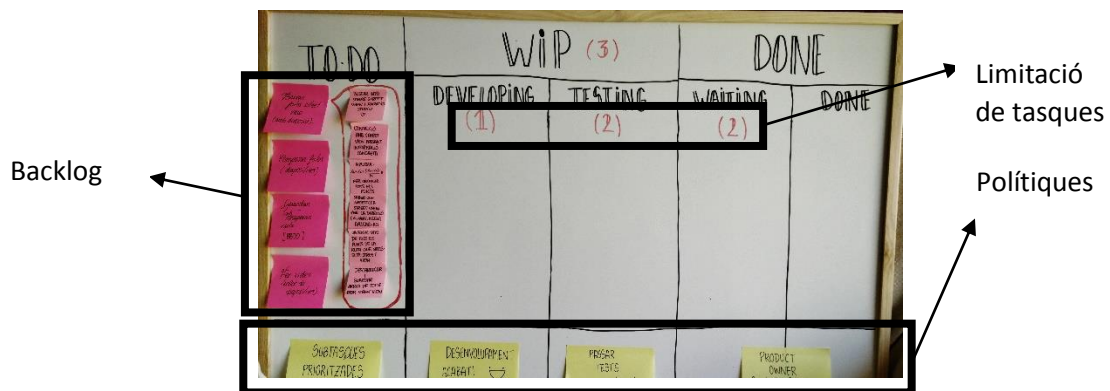


Figura 23. Taulell que s'ha realitzat per visualitzar el flux de treball i aplicar Kanban.

A la figura 23 podem veure la primera història d'usuari dividida en les subtasques, que són les que es desplaçaran al llarg del taulell i la senyalització de les principals característiques del taulell de Kanban, la visualització de backlog, la limitació de tasques i les polítiques.

Es van realitzar fotografies de l'evolució del taulell Kanban, per tal de poder apreciar la tècnica de "pull don't push".



Figura 24. Evolució del taulell Kanban.

Les figures 24.a – 24.f ens mostren la primera part de la representació del flux de treball de la primera història d'usuari per a la subtasca de visualització d'imatges i recuperació de rutes aplicant la metodologia de gestió Kanban.

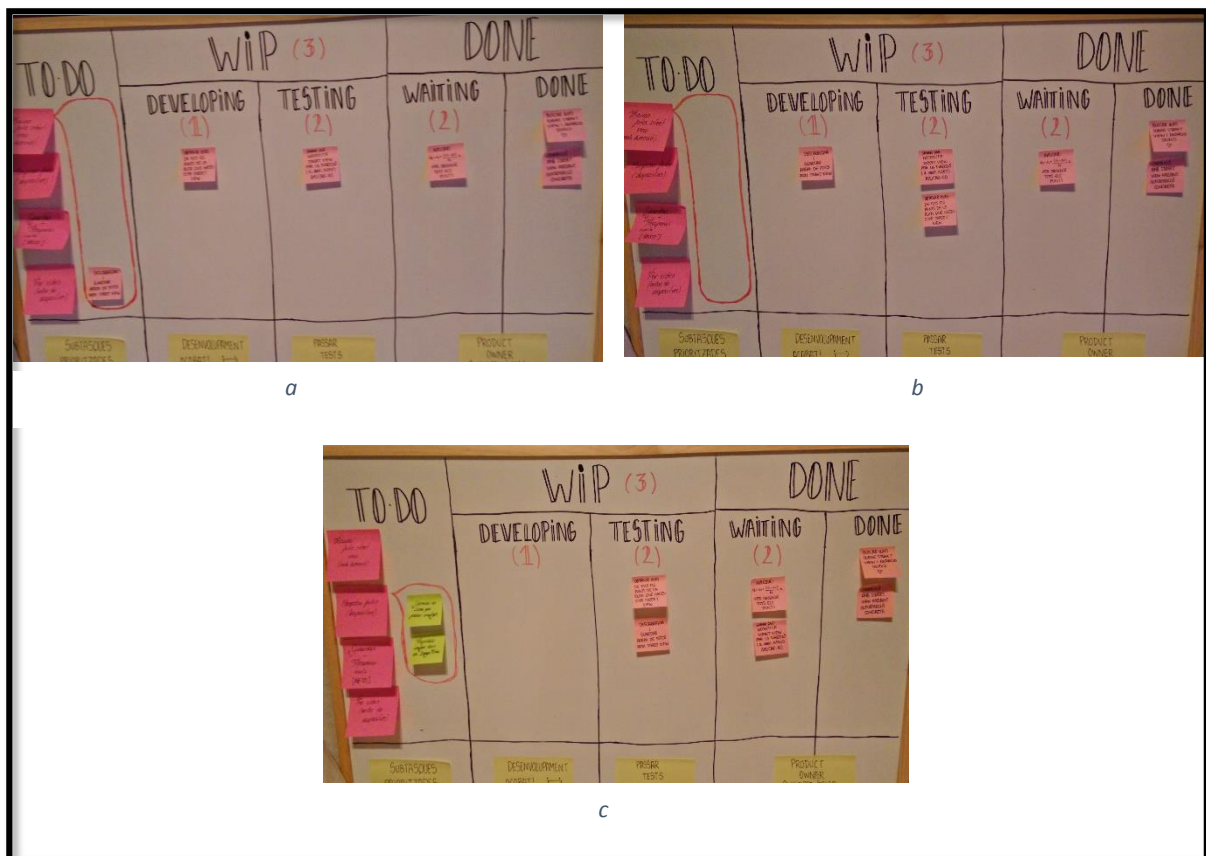


Figura 25. Evolució del taulell Kanban.

Les figures 25.a – 25.c ens mostren la finalització del flux de treball de la primera història d'usuari per a la subtasca de visualització d'imatges i recuperació de rutes aplicant la metodologia de gestió Kanban.

Tal i com es pot veure a les fotografies, s'ha respectat el WIP de totes les columnes, i al finalitzar la primera setmana, s'havia realitzat tot el desenvolupament, només quedava testejar i obtenir l'acceptació del product owner.

Un cop acabada aquesta setmana ja es descarregaven correctament les imatges de Street View, mitjançant una tasca asíncrona que descarrega la imatge que obté de la URL creada, amb la imatge de Street View. Aquestes URLs que contenen les imatges són del tipus:

- <http://maps.googleapis.com/maps/api/streetview?size=400x400&location=latitud,lonitud&fov=90&heading=heading>

El atributs mínims requerits són la longitud, la latitud i la mida de la finestra on es mostra la imatge. Els atributs opcionals són, en aquest cas, fov i heading, però se'n poden afegir més.

Els atributs ressaltats en negreta, latitud, longitud i heading es calculen per cada imatge. És a dir, primer obtenim els punts equidistants de la ruta, i després obtenim la latitud i longitud del punt, i calculem el heading segons el següent punt, que ens indica la direcció.

L'atribut fov determina el camp de visió horitzontal de la imatge. El camp de visió s'expressa en graus, amb un valor màxim de 120. Quan es tracta d'una finestra de mida fixa, com ara una imatge de Street View d'una grandària fixa, representa el zoom; com més baix és el nombre, més elevat és el zoom. Per defecte és 90, i és el valor que s'ha deixat.

L'atribut heading indica la direcció de la brúixola de la càmera. Els valors acceptats van de 0 a 360 (0 indica Nord, 90 indica Est, 180 indica Oest i 360 Sud). Si no s'especifica cap valor, el valor per defecte dirigirà la càmera des de la localització especificada cap al punt on es va fer la fotografia més propera. S'ha hagut d'especificar, de manera que segueixi la direcció de la ruta que ha escollit l'usuari.

A nivell de desenvolupament, per cada punt de la ruta descarreguem en un thread secundari la seva imatge, i la guardem a una array, per tal de no haver de descarregar totes les imatges cada cop.

El principal inconvenient d'aquest mètode és que Google Street View fixa un límit de fotografies descarregades dins de l'aplicació, a no ser que es pagui pel servei, quan s'excedeix el límit d'imatges descarregades l'objecte InputStream no pot descarregar les imatges i per tant la ruta es veu tallada.

La següent setmana es va centrar en la composició de les imatges per simular un vídeo, la tasca principal és:

- Com a usuari vull que, tot i que siguin fotos, estiguin composades de manera que doni la sensació de que és un vídeo.

Les subtasques que es van col·locar al backlog són:

1. Generar un timer per passar les imatges
2. Reproduir imatges dins de l'objecte Image View
 - a. Un objecte Image View és un component que ens permet visualitzar qualsevol tipus d'imatge.

L'evolució del taulell durant la setmana va ser el següent:

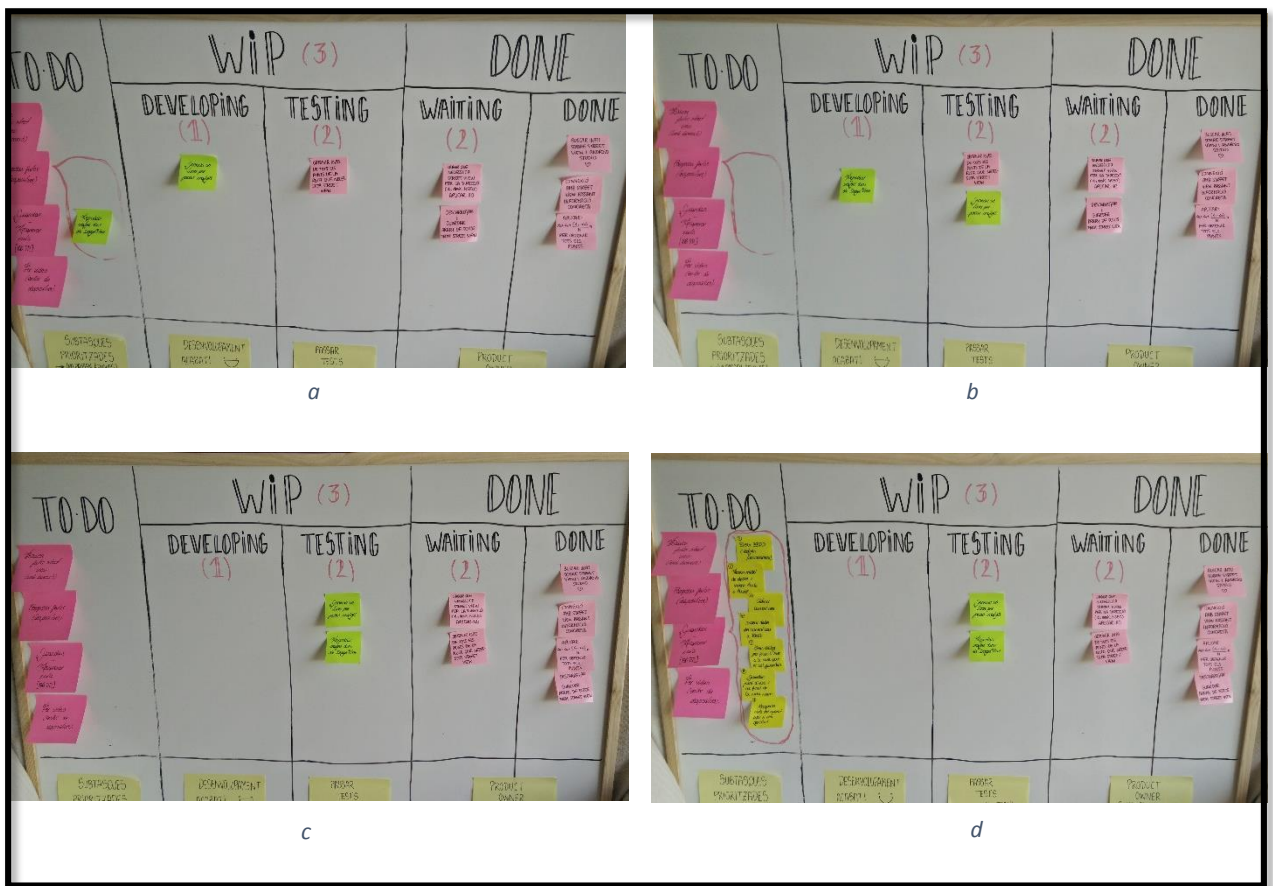


Figura 26. Evolució del taulell Kanban.

Les figures 26.a – 26.d ens mostren la representació del flux de treball de la segona història d'usuari per a la subtasca de visualització d'imatges i recuperació de rutes aplicant la metodologia de gestió Kanban.

Com es pot observar, es van acabar de testejar tasques de la setmana anterior i es va acabar el desenvolupament per compondre les imatges de manera que simulessin un vídeo.

Un cop acabada la setmana, s'havia creat un timer que reproduïa les imatges guardades en un array dins del Image View de l'activity.

Finalment, durant la tercera setmana es va realitzar la història d'usuari que quedava:

- Com a usuari vull guardar i recuperar les rutes que he creat.

D'aquesta història d'usuari es van crear les següents subtasques:

1. Crear base de dades i testejar funcionament.

2. Model de dades i inserció de les taules.
3. Obtenir current user.
4. Inserir dades del current user a base de dades.
5. Implementar funcionalitat del dialog per ficar-li nom a la ruta i guardar-la.
6. Guardar punt inicial i punt final de la ruta i el nom.
7. Recuperar rutes del current user si està registrat.

L'evolució del taulell d'aquesta setmana, que inclou el desenvolupament de les subtasques enumerades anteriorment i la finalització de la subaplicació és la següent:

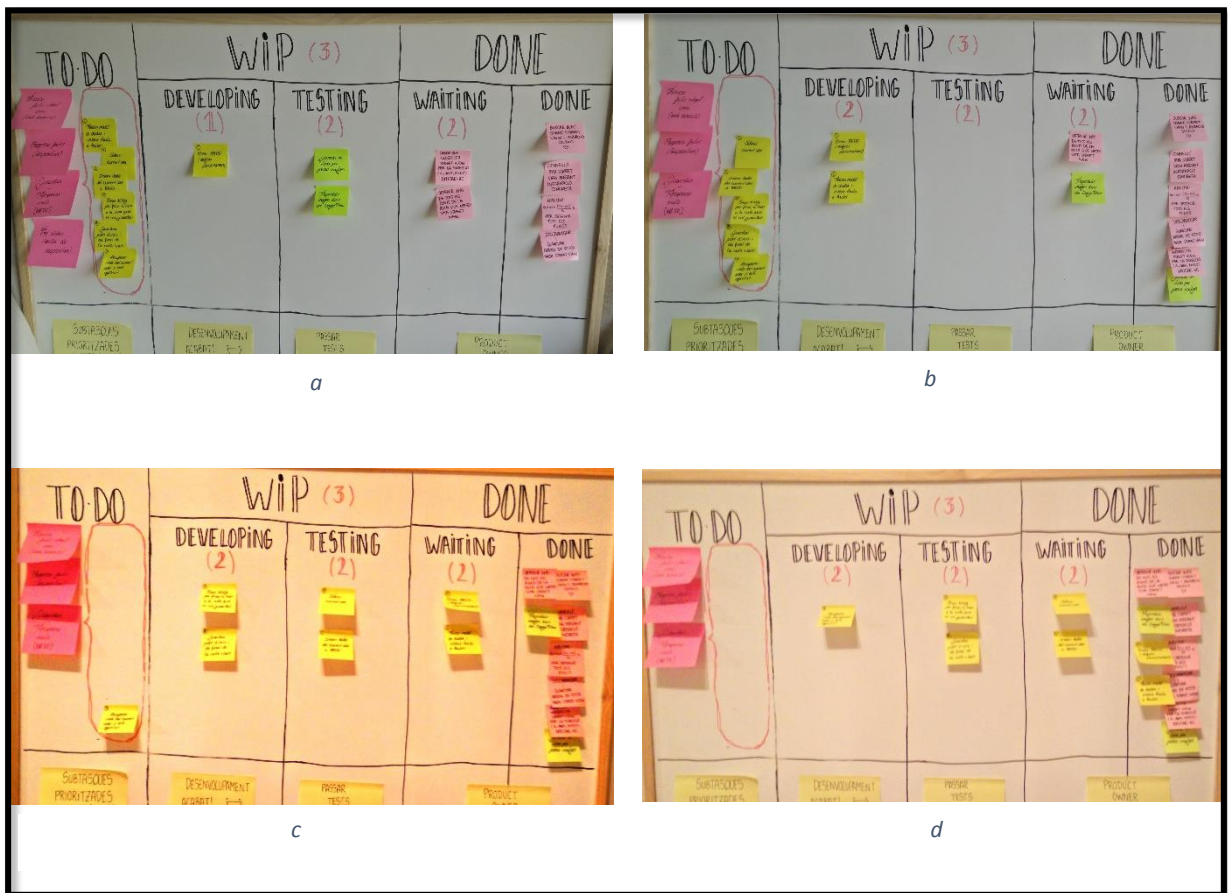
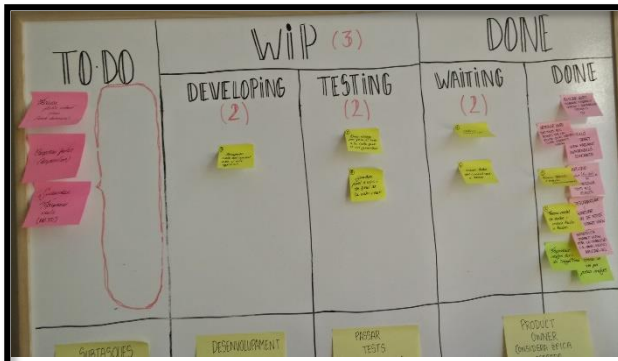


Figura 27. Evolució del taulell Kanban.

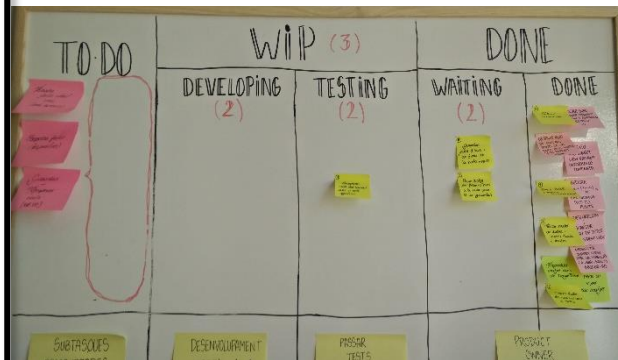
Les figures 27.a – 27.d ens mostren la primera part de la representació del flux de treball de la tercera història d'usuari



a



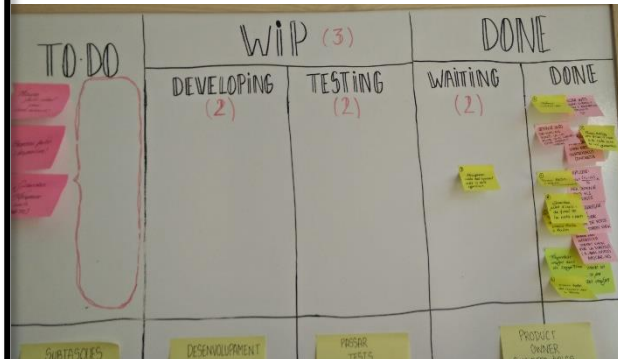
b



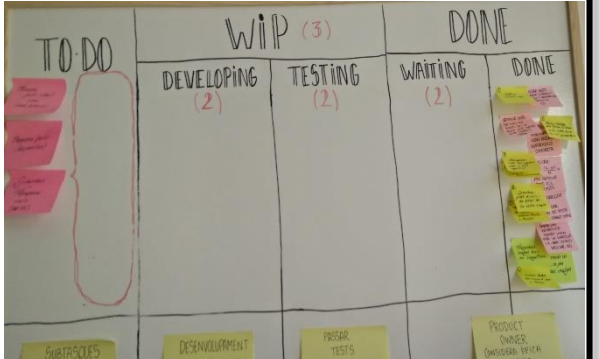
c



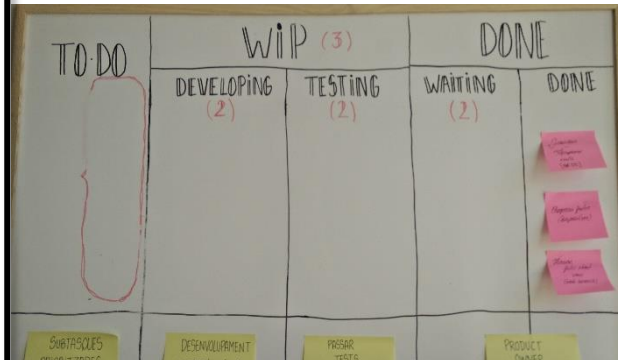
d



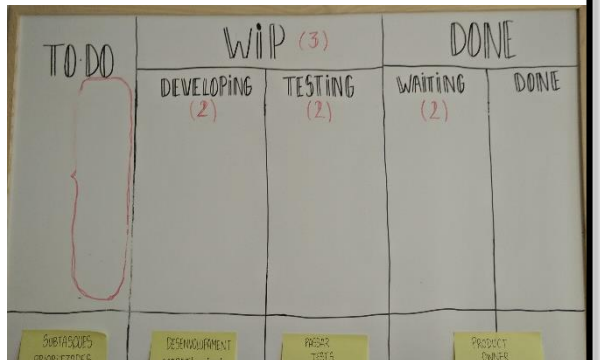
e



f



g



h

Figura 28. Evolució del taulell Kanban.

Les figures 28.a – 28.h ens mostren la finalització del flux de treball per a la tercer subtasca de la visualització i recuperació de rutes i tancament de la subaplicació.

Com es pot observar, durant el desenvolupament d'aquesta última història d'usuari, es va haver d'augmentar el nombre de tasques a la columna de desenvolupament a 2, degut a la formació d'un coll d'ampolla, com es pot veure a la figura 27, i més concretament a la figura 27.c.

El coll d'ampolla es va generar degut a una acceptació de les tasques massa tardana, fet que impossibilitava avançar amb el desenvolupament.

Un cop acabades les subtasques de la història d'usuari, es va procedir a l'acceptació de totes les tasques èpiques i la finalització de la subaplicació.

4.2.2 Característiques aplicades

Un cop finalitzada la subaplicació de la visualització i recuperació de rutes utilitzant la metodologia de gestió Kanban, podem observar que s'han aplicat totes les característiques d'aquesta metodologia correctament.

Primerament, la més representativa; la visualització del flux de treball. Com s'ha pogut demostrar amb les imatges presentades anteriorment, durant tot el procés de desenvolupament s'ha pogut visualitzar el flux i evolució del procés.

S'han redactat les històries èpiques i les subtasques en post-its que s'han col·locat al backlog (TO DO) de la pissarra, posteriorment, les subtasques s'han anat desplaçant al llarg del taulell, dividit en les columnes necessàries, que representen els estats del procés de desenvolupament, i s'ha pogut veure com avancen i es finalitzen les diferents tasques del backlog fins quedar-se vuit.

Seguidament, també s'ha limitat el treball en curs, és a dir, el WIP, evitant, d'aquesta manera, la saturació i mal desenvolupament de les tasques per overflow de treball.

La limitació del WIP ha sigut útil per organitzar correctament les tasques i acabar correctament totes les característiques de cadascuna abans de començar la següent.

També s'han redactat les polítiques d'acceptació i s'han col·locat a la pissarra per fer-les visibles. Les polítiques redactades han servit per assegurar la qualitat i usabilitat de la subaplicació, ja que han obligat a cadascuna de les tasques a complir una sèrie de normes, com superar la fase de test i obtenir l'aprovació del product owner.

Finalment, s'ha aplicat correctament la metodologia "pull don't push" pel desplaçament de les targetes dins de la pissarra. Segons el WIP establert en cada columna, s'han anat ficant noves tasques quan les anteriors complien les polítiques i podien passar a la següent columna.

Totes aquestes característiques aplicades es poden veure representades a les imatges del punt 4.2.1, i aquesta és la principal motivació que ofereix Kanban, que en una sola pissarra es pot veure el funcionament del procés de producció de software i l'evolució del projecte, únicament mitjançant post-its i una pissarra dividida en columnes.

4.2.3 *Conclusions personals*

Un cop analitzada l'aplicació de Kanban per a la subaplicació de visualització i recuperació de rutes, s'han extret punts positius i negatius de la metodologia.

A nivell de punts positius, crec que la visualització del flux de treball és un factor fonamental per a qualsevol metodologia de gestió de projectes. No només ofereix una plataforma magnífica per organitzar el treball visualment i no haver de fer memòria constantment, sinó que a més a més, motiva i fa més eficient al programador, ja que només el fet de poder moure una tarja d'una columna a una altra ja és un incentiu importantíssim, és una representació física de l'esforç aplicat al procés de producció.

També considero un punt molt positiu la prioritització de tasques del backlog, com en el cas d'Scrum, ja que permet optimitzar el temps que es dedica i no malgastarlo realitzant tasques sense prioritat.

En el cas concret d'aquest projecte, s'havia plantejat la idea de crear un vídeo de la ruta en comptes de reproduir les imatges mitjançant un timer; finalment es va descartar per, primerament, ser innecessari, ja que el timer complia els requisits d'usuari correctament, i després per considerar que era una tasca sense valor de negoci i que generava una pèrdua de temps important.

En el cas de que no s'hagués col·locat aquesta tasca com a última prioritat, probablement s'hagués desenvolupat en primera instància, quan realment era més complicat que la creació d'un timer i completament innecessari.

Per tant, si en una aplicació amb una sola persona en el procés de desenvolupament, Kanban ja ha suposat una optimització de temps, en grans equips amb gran projectes de llarga durada,

suposa una millora de rendiment més que remarcable a nivell d'entrega de software que compleix les característiques que espera el client, sense generar més del que és necessari i assegurant la qualitat del producte.

Un altre punt que considero molt positiu de Kanban és el dinamisme que implica treballar amb una pissarra que et recorda constantment el que has de fer, però que també implica que t'aixequis i canviïs les targetes de lloc, aporta més optimisme durant el procés de desenvolupament i un dinamisme poc usual en el processos de desenvolupament de software.

Finalment, un altre punt positiu molt important, és la gran ajuda a nivell d'aclariment de conceptes que aporta Kanban. El fet de veure representades les històries d'usuari i les subtasques en post-its ajuda a poder ordenar correctament els objectius de l'aplicació i tenir una visió de futur del projecte, podent-t'ho veure constantment representat en una pissarra.

A nivell de punts negatius, l'únic que considero que és remarcable en el cas de Kanban és la poca limitació en el timing d'entrega. En origen, no em sembla una mala idea donar llibertat de dates als desenvolupadors, ja que crec que així el rendiment pot ser major, però el principal inconvenient de no tenir un timing fixe al que adaptar-se, és que si no s'adopta la filosofia de mantenir un ritme constant de treball, donant sempre el màxim d'esforç i qualitat en el software desenvolupat, és complicat que Kanban pugui assegurar una entrega a temps.

Crec que li falta, no la modificació de la filosofia Kanban, sinó un altre model, com ara Scrum, que porti una base sòlida a nivell d'organització temporal del procés de creació del projecte.

Tot i així, com ja he comentat, si els desenvolupadors son capaços d'assumir la filosofia que implica Kanban, és a dir, donar sempre el màxim i el millor d'un mateix a l'hora de desenvolupar software, és una filosofia que aporta una llibertat molt interessant als treballadors, de manera que també dona més seguretat a aquests, ja que s'ha de depositar molta confiança en el treballador per a que compleixi el timing, i això aporta motivació.

En conclusió, crec que Kanban és una metodologia de gestió de projectes que es en la seva totalitat positiva si els treballadors que l'utilitzen són capaços d'entendre i assumir la filosofia de transfons que implica.

La visualització del flux de treball, i la representació en una acció física de l'esforç que s'ha dedicat em semblen una gran motivació. La representació de tasques en targetes i dels estats del procés en columnes és una gran manera de tenir un mapa del procés sense haver-lo de reproduir mentalment cada vegada.

4.2.4 Resultats

Els resultats obtinguts un cop finalitzada l'aplicació de Kanban han sigut els següents.

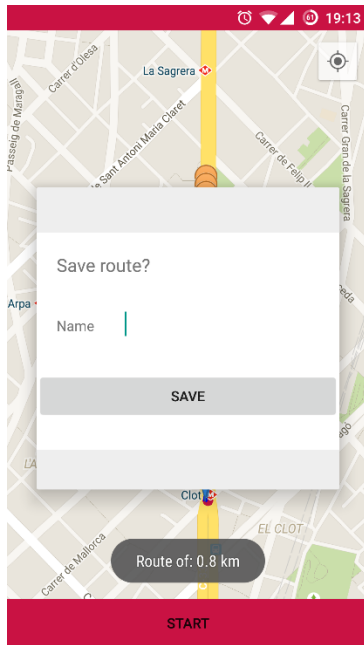


Figura 29. Dialog amb funcionalitat implementada

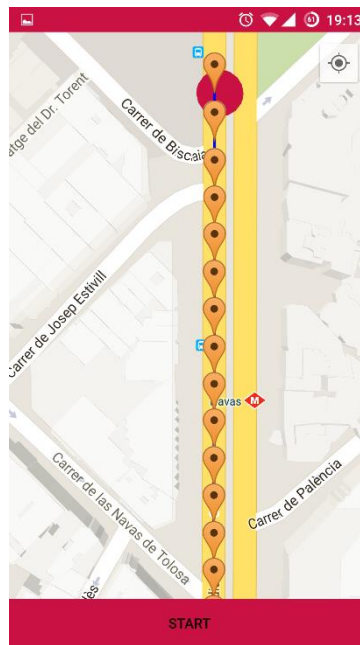


Figura 30. Obtenció de punts equidistants per Street View

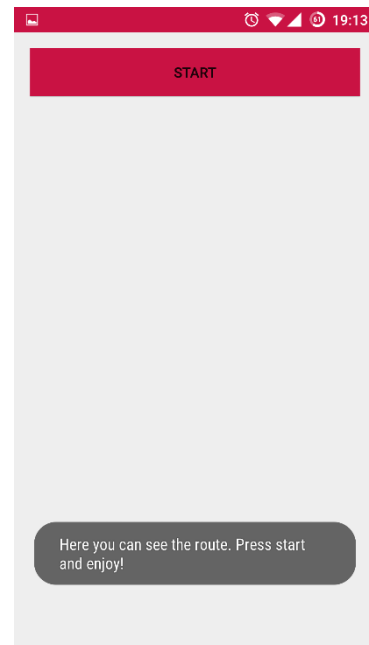


Figura 31. Activity per visualitzar la ruta

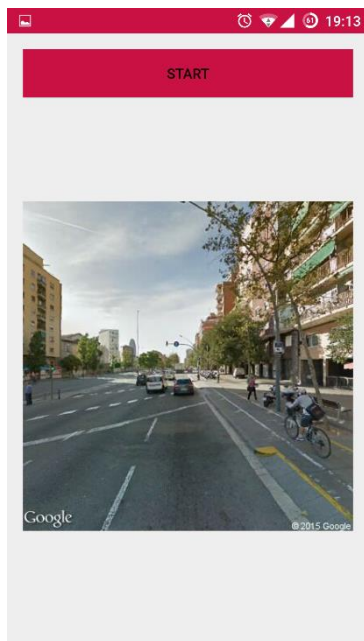


Figura 32. Visualització de la ruta

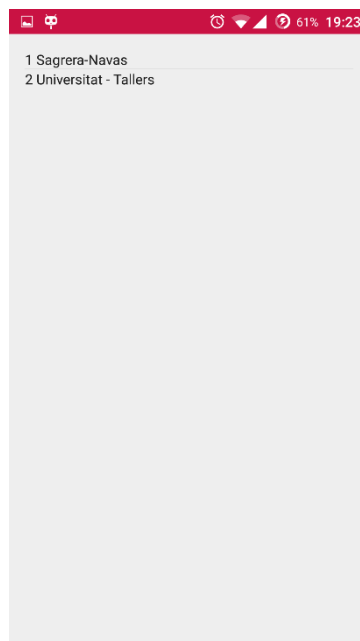


Figura 33. Recuperació de rutes creades

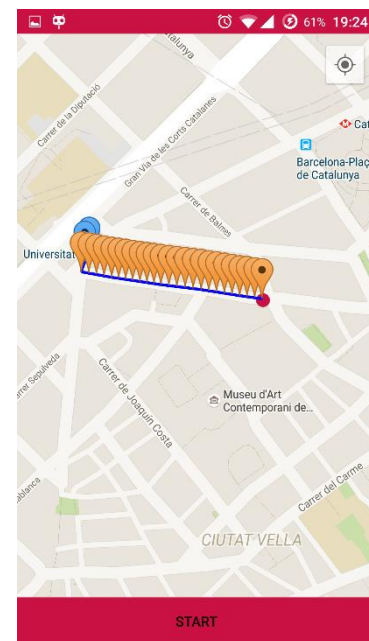


Figura 34. Visualització de la ruta recuperada Universitat - Tallers

Les figures 29 i 30 fan referència a les implementacions fetes dins l'Activity de selecció de ruta, és a dir, el guardar la ruta a base de dades i l'obtenció de punts equidistants per desaccregar les imatges. Les figures 31 i 32 ens mostren la previsualització de la ruta, la 33 i la 34 fan referència a la recuperació de rutes de base de dades.

A nivell de codi, per la realització d'aquesta subaplicació s'han creat i modificat les següents classes que es poden visualitzar a l'annex número 1 d'aquesta memòria:

- Vista
 - VisualizacioRuta.java
 - Mostra la interfície per visualitzar la ruta, amb un botó START
 - Un cop seleccionem start, si hi ha una ruta seleccionada, ja sigui recuperada de base de dades o recentment creada, es calcula la informació necessària de cada punt de la ruta per introduir-ho a la url de Street View.
 - Es calcula el heading entre les imatges (s'ha hagut de calcular el bearing i ficar-ho al heading, ja que a nivell de url no es possible ficar el heading, però el nostre bearing és el heading que espera Street View).
 - S'obté la latitud i longitud de cada punt .
 - Un cop obtinguda la url s'inicia una tasca asíncrona que descarrega les imatges mitjançant un objecte de tipus InputStream.
 - Es descarreguen i guarden les imatges en un array de tipus Bitmap.
 - Només es descarreguen les imatges utilitzant internet el primer cop.
 - Es reproduïxen les imatges dins del Image View.
 - PathGoogleMapActivity.java
 - Modificació de la funcionalitat del dialog perquè guardi les dades de la ruta i de l'usuari a base de dades.
 - S'han creat tots els xml corresponents a les activies declarades.
- Controlador
 - Controlador.java
 - Modificació i creació de funcions per enllaçar les noves funcionalitats de la vista amb el model.
- Model

- InformacionRuta.java
 - Conté l'array de punts de la ruta, un boolean que indica si hi ha una ruta seleccionada o no, el bearing, indica si es una ruta recuperada de base de dades o no mitjançant un boolean, guarda el punt inicial i el punt final de la ruta.
 - S'utilitza com a classe compartida en temps d'execució de l'aplicació. Ja que les activitats depenen entre elles, és necessària una classe compartida entre elles.
- InformacionUsuario.java
 - Classe compartida en temps d'execució.
 - Conté la informació de l'usuari que es necessita compartir entre activitats.
- ReproducirImagenes.java
 - Crea el timer per visualitzar les imatges simulant un vídeo dins del Image View.
- SoporteSelectRuta.java
 - Obté els punts equidistants de la ruta, dels quals descarreguem les imatges de Street View.
- Usuario.java
 - Conté tota la informació de l'usuari.
 - Mail
 - Identificador
 - Si ha vist o no el tutorial.
 - La part del tutorial s'explicarà a l'apartat 4.3 - Lean.
- Ruta.java
 - Conté tota la informació de la ruta.
 - Nom
 - Punt inicial i final
 - Usuari que l'ha creat
 - Identificador
- MySQLiteHelper.java
 - Crea la base de dades i les taules necessàries
 - El model de dades crea és el següent

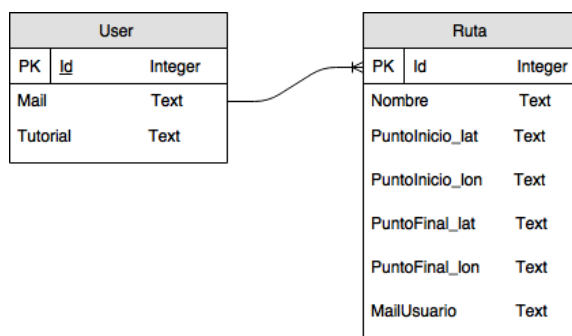


Figura 35. Model de dades.

La figura 35 ens mostra el model de dades, on un usuari pot crear n rutes. La taula usuari conté els camps e-mail de l'usuari i tutorial, que indica si l'usuari ha vist o no el tutorial, i la seva clau primària és un identificador de tipus Integer. La taula ruta, amb una clau primària tipus Integer, conté els camps referents als nom de la ruta, el punt inicial i final separats en latitud i longitud de la ruta, i el e-mail de l'usuari que ha creat la ruta.

4.3 Lean

4.3.1 Disseny de la FASE 3

L'última part de l'aplicació, l'opció d'escoltar música pròpia o de la gent del voltant, mitjançant un reproductor de música integrat dins de l'aplicació, ha sigut l'escollida per realitzar aplicant la metodologia de gestió Lean.

A part, també es va realitzar una millora general de l'aplicació a nivell d'interfície, creació d'un tutorial al iniciar l'aplicació i qüestions d'usabilitat, com ara missatges explicatius, icones per seleccionar les subaplicacions des de qualsevol punt de l'aplicació etc.

La idea general d'aquesta tercera subaplicació és que l'usuari tingui l'opció de poder escoltar la música de la gent que l'envolta, de manera que la primera cançó que li apareix a la llista sigui la de la persona més propera. Per tal de desenvolupar-ho a nivell local i poder testejar el funcionament tenint en compte la localització de l'usuari, s'han creat usuaris ficticis a la base de dades, col·locats en diferents punts de Barcelona (Plaça de Glòries, Plaça Universitat, Plaça Catalunya, centre de Sant Cugat, centre de Sabadell, Rambles, Horta i a la Barceloneta), i se'ls hi ha assignat una cançó; d'aquesta manera s'ha simulat el servidor on s'allotjaria la base de dades en cas de que l'aplicació sortís al mercat.

Per tal de realitzar Lean, al ser més aviat una filosofia sense unes pautes massa clares, es va decidir aplicar l'eliminació dels Seven Wastes of Lean en combinació amb Kanban per visualitzar el flux de treball.

Lean ha resultat ser la part més complicada, ja que, com s'ha dit, no hi ha un "manual d'ús" com en el cas de Kanban i Scrum; és més aviat una metodologia de gestió de projectes que ens dona les eines perquè el projecte tingui èxit un cop surti al mercat, però no ens dona l'opció de realitzar un seguiment del procés de desenvolupament.

A continuació es presentaran els Seven Wastes of Lean i com s'han adaptat en el procés de desenvolupament de l'aplicació, és a dir, quines son les tècniques que es faran servir per eliminar-los.

1. **Reduir o limitar WIP:** s'utilitzarà la limitació del WIP del taulell de Kanban.
2. **No produir més del que es demana:** excel·lent prioritització de tasques i creació de subtasques ben especificades.
3. **Evitar més passos dels necessaris en els processos:** fer el codi més simple possible, es mesurarà en primera instància en línies de codi, i seguidament a nivell d'organització i estructura de codi.
4. **Passar correctament feina d'una persona a l'altra:** es realitzarà una user story d'entrada i de sortida. La user story de sortida seran les modificacions a nivell de codi que s'han fet, de manera que si alguna altra persona hagués de modificar el codi, fos conscient de les classes i funcionalitats modificades i afegides.
5. **Evitar endarreriment:** es seleccionaran les tasques segons valor de negoci, donant sempre prioritat a aquelles tasques que aportin valor per a l'usuari final.
6. **Evitar canvi de tasques:** es tindrà sempre visible el taulell de Kanban amb les tasques priorititzades, de manera que no s'anirà canviant de tasca, es realitzaran les escollides del backlog.
7. **Evitar defectes:** redacció d'històries d'usuari INVEST, amb un criteri d'acceptació detallat per cada història. Una història d'usuari INVEST compleix els següents punts:
 1. **INDEPENDENT**
 - Les tasques d'usuari han de ser tant independents entre elles com sigui possible, de manera que el desenvolupament o no desenvolupament d'una no afecti a una altra.
 2. **NEGOCIABLE**

- S'ha de poder discutir sobre les històries d'usuari, i s'han de poder modificar al llarg del procés de desenvolupament; no es tracta de prendre decisions en una primera instància, sinó de definir les idees principals que poden anar modificant-se durant el procés de creació del producte.

3. VALUABLE

- Ha de ser avaluable, de manera que sigui possible veure si compleix o no tots els requisits estipulats per a considerar-se un producte preparat per presentar al client.

4. ESTIMABLE

- Ha de ser estimable, és a dir, s'ha de poder estimar el temps i el cost de cada tasca, així com el seu valor de negoci, per tal de realitzar una bona prioritització del backlog.

5. SMALL

- Han de ser curtes. Amb històries èpiques d'usuari la flexibilitat de cara als errors i modificacions es pràcticament nul·la; en canvi, desenvolupar tasques curtes fa que la gestió d'errors sigui més assequible i es puguin fer canvis. A part, també ajuda a que les tasques siguin independents entre elles.

6. TESTEJABLE

- S'ha de poder testejar un cop finalitzada la tasca, per considerar-la vàlida o no.

Un cop definits els Seven Wastes of Lean, es va definir la història d'usuari èpica:

- Com a usuari, vull un reproductor de música que em permeti escoltar la música que tinc al meu dispositiu, que es comparteixi automàticament amb els altres usuaris i poder escoltar el que escolta la gent al meu voltant

Un cop definida la història d'usuari èpica, com en els altres casos, es van definir les següents subtasques:

- Com a usuari vull un reproductor de música.
 - Crear reproductor de música amb interfície intuïtiva.

- Que el reproductor de música mostri les cançons del dispositiu.
- Poder reproduir les cançons.
- Poder parar/pausar la cançó.
- Passar a la següent cançó .
- Com a programador necessito simular un servidor.
 - Crear taula a la base de dades que guardi: cançó, usuari i posició.
 - Guardar a la base de dades la cançó i la posició de l'usuari.
 - Tenir una base de dades amb dummy data que simuli altres usuaris amb les seves posicions.
- Com a usuari vull tenir l'opció d'escoltar la meua música o la dels demés.
 - Crear dialog que permeti escollir una opció o una altra.
 - Carregar una llista de cançons o una altra segons l'opció escollida.
- Com a usuari vull que les cançons de les persones més properes a mi apareguin les primeres de la llista.
 - Obtenir current position de l'usuari i calcular la distància amb tots els altres usuaris de la base de dades.
 - Evidentment, si l'aplicació no simulés el servidor, es limitaria el radi per mostrar les cançons dels altres usuaris.

Un cop definides les tasques, es va utilitzar la metodologia Kanban per visualitzar el flux de treball.

A continuació es mostra el taulell Kanban preparat per iniciar el desenvolupament de les tres subtasques principals:

1. Reproductor de música.
2. Simulació del servidor.
3. Escoltar música dels demés.

Les subtasques escollides per a la primera setmana de desenvolupament són:

1. Mostrar cançons del dispositiu.
2. Botons play, pause, avançar i anar enrere en la cançó, cançó següent i cançó anterior.
3. Interfície intuïtiva.

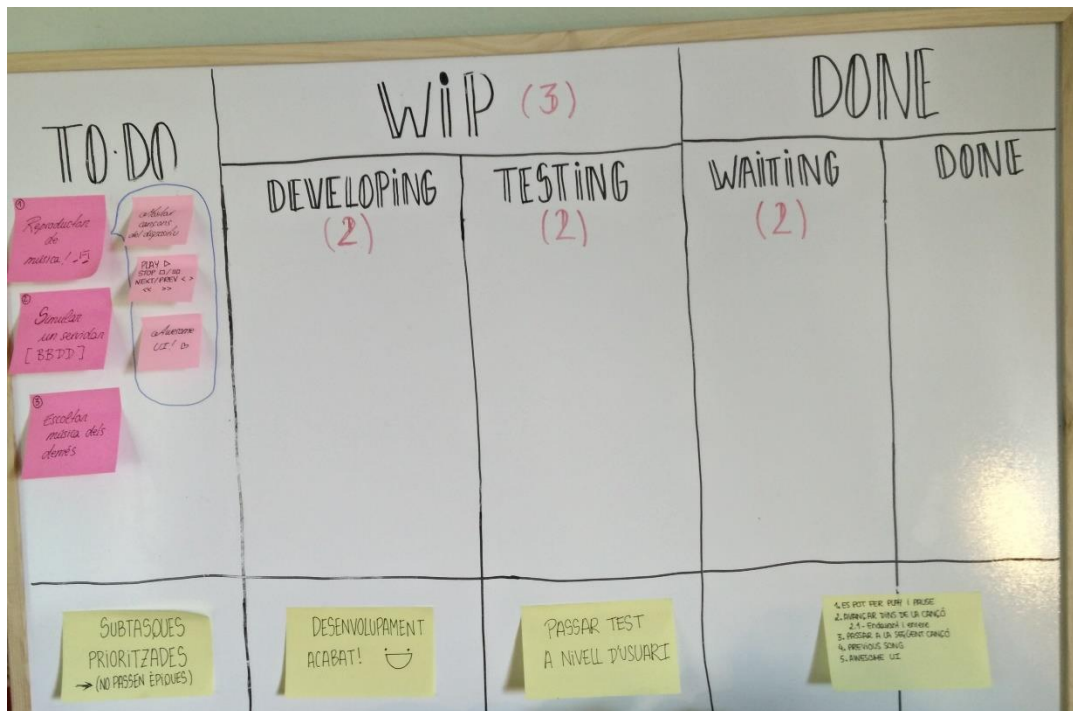


Figura 36. Taulell preparat per iniciar el desenvolupament.

La figura 36 ens mostra el taulell de Kanban preparat per iniciar el desenvolupament aplicant l'eliminació dels Seven Wastes of Lean. Podem veure el backlog, les polítiques i la limitació del WIP.

Com es pot observar al taulell, s'ha limitat el WIP en les columnes necessàries, i les polítiques d'acceptació s'han detallat.

Per tal de tenir unes polítiques d'acceptació concretes, s'han definit específicament per a cada tasca; de cara a la primera tasca, les polítiques d'acceptació són les següents:

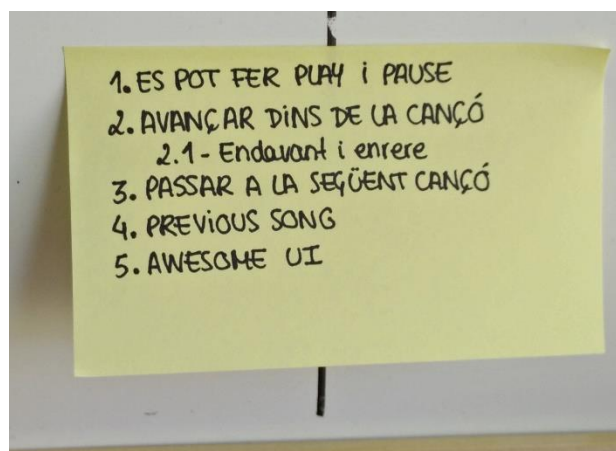


Figura 37. Polítiques d'acceptació necessàries a complir per passar de waiting a done i considerar la tasca del reproductor de música finalitzada.

La figura 37 ens mostra les polítiques necessàries a complir per passar de waiting a done, que són:

1. Es pot fer play i pause.
2. Avançar dins de la cançó, enfavant i enrere.
3. Passar a la següent cançó.
4. Passar a la cançó anterior.
5. Interfície d'usuari intuïtiva.

Un cop finalitzades les tasques, tal i com s'indica al Waste número quatre, passar correctament feina d'una persona a l'altra, es va realitzar una tarja explicativa amb les modificacions efectuades:

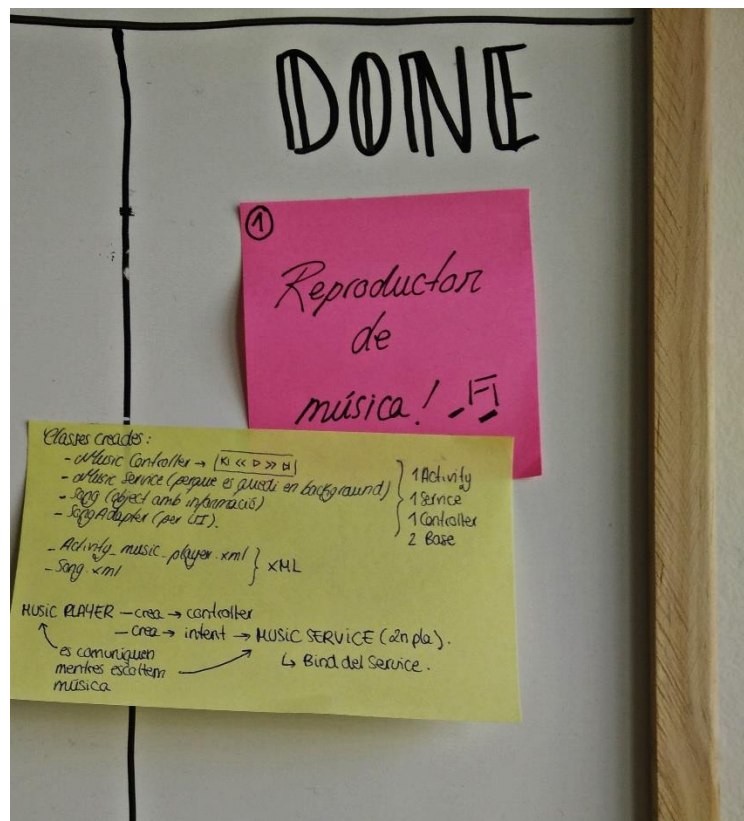


Figura 38. User story de finalització de la tasca.

La figura 38 ens mostra la tarja realitzada al finalitzar la tasca del reproductor de música. Aquesta tècnica fa referència al waste número quatre dels Seven Wastes of Lean.

De cara a les següents tasques, es va decidir ajuntar-les per optimitzar el temps i realitzar una única tarja d'acceptació.

Les tasques definides són:

1. Millorar interfície.
2. Dialog per escollir escoltar la música del dispositiu o de les persones del voltant.
3. Crear taula a la base de dades per allotjar la informació necessària.
4. Dummy data per a usuaris ficticis dins de la base de dades.
5. Carregar un llistat de cançons o un altre segons l'opció escollida per l'usuari.
6. Obtenir les n primeres cançons dels altres usuaris ordenades segons proximitat.

La tarja referent a les polítiques d'acceptació és la següent:

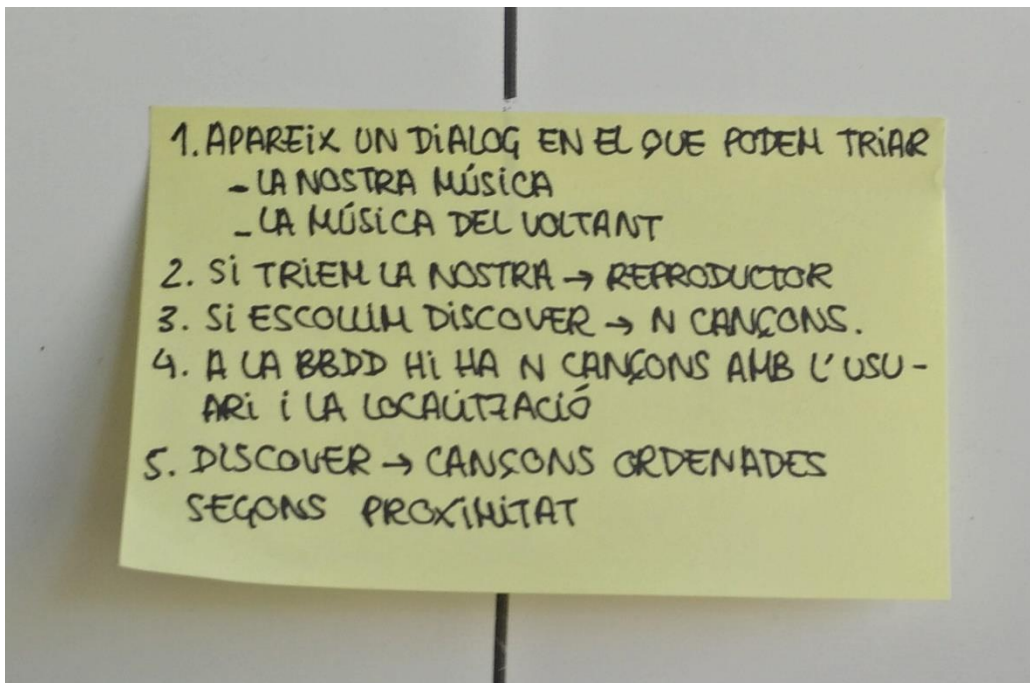
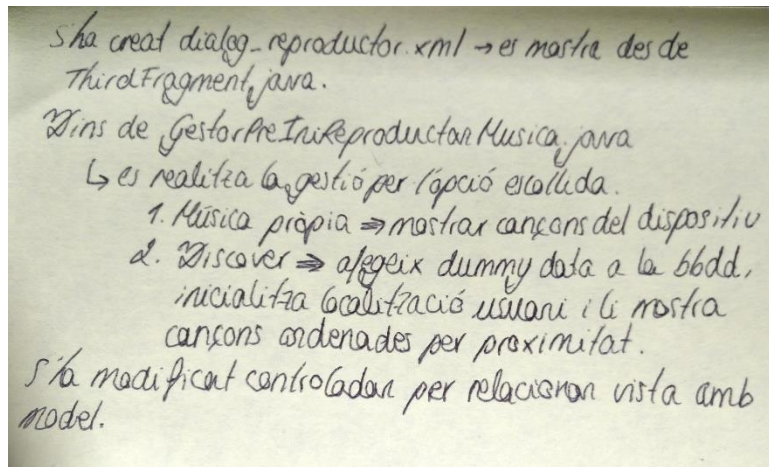


Figura 39. Polítiques d'acceptació per a les tasques de simulació del servidor i escoltar la música dels demés.

La figura 39 ens mostra les polítiques necessàries a complir per passar de waiting a done, que són:

1. Dialog per seleccionar llista de reproducció.
2. Si triem la nostra música passem al reproductor.
3. Si escollim discover passem a veure el llistat ordenat de cançons.
4. A base de dades hi ha n cançons amb l'usuari i la localització.
5. El llistat ordenat de cançons s'ordena segons distància.

La tarja resum de les principals modificacions fetes durant aquesta tasca és:



Sha creat dialog_reproductor.xml → es mostra des de ThirdFragment.java.
Dins de GestorPreTruReproductorMusica.java
↳ es realitza la gestió per l'opció escollida.
1. Música pròpia ⇒ mostrar cançons del dispositiu
2. Discover ⇒ afgeix dummy data a la bbdd, inicialitza localització usuari i li mostra cançons ordenades per proximitat.
S'ha modificat controlador per relacionar vista amb model.

Figura 40. User story creada al finalitzar la tasca.

La figura 40 ens mostra la tarja resum de les principals modificacions que s'han fet durant el desenvolupament de les tasques dos i tres del backlog.

Un cop finalitzades totes les tasques mínimes necessàries per considerar que el producte complia tots els requeriments, es van realitzar altres modificacions, com ara la creació d'un tutorial el primer cop que s'inicia l'aplicació. Un cop acabat el tutorial, l'usuari té l'opció de no tornar-lo a veure.

Després, es van millorar aspectes de la interfície, es van afegir una sèrie d'icones a la finestra principal, de manera que es pogués accedir a les diferents subaplicacions a través del menú desplegable i a través de les icones.

A part, es van millorar els missatges que es mostraven en cada subactivitat, de manera que l'experiència d'usuari fos més simple.

Finalment, es va optimitzar el codi i es va dividir en packages per seguir el patró MVC (Model Vista Controlador) correctament i aplicant la solució per al Waste número tres.

4.3.2 *Característiques aplicades*

La principal característica de Lean aplicada per al desenvolupament de la última subaplicació de ROAD han sigut l'eliminació dels Seven Wastes of Lean.

Després d'analitzar-los amb profunditat va ser possible adaptar-los a la realitat del projecte que s'estava desenvolupant, i es va buscar la manera d'eliminar el malbaratament amb diferents tècniques, com ara la limitació del WIP, la prioritització correcta de les subtasques, subtasques INVEST molt detallades i polítiques d'acceptació específiques per a cada tasca.

Va resultar impossible definir la idea de negoci mitjançant el Lean Canvas, ja que no era realista ni possible definir tota la part referent guanyos econòmics, i per tant, s'hagués deixat el Canvas a mig omplir.

Tot i així, aplicant la filosofia de reduir al màxim el malbaratament de recursos mitjançant els Seven Wastes of Lean, ja es compleix la principal base de la filosofia Lean, que és reduir el malbaratament i produir només allò que té valor de negoci i és necessari.

A part, també es va aplicar al prioritització de tasques segons el valor de negoci, un altre factor clau en la filosofia Lean, que s'orienta completament als futurs beneficis del projecte i a mantenir el client satisfet.

4.3.3 *Conclusions personals*

Després d'aplicar Lean per al desenvolupament de la última subaplicació, s'han detectat punts forts i punts febles d'aquesta metodologia.

A nivell de punts forts, s'ha de dir que Lean és una filosofia que optimitza el producte, per exemple, d'una empresa que acaba de néixer, les anomenades Start Ups, ja que obliga a fer un anàlisi del mercat i de l'avantatge competitiva en relació a altres producte. A part, la reducció del malbaratament durant el procés de desenvolupament de software fa que tot el procés sigui més àgil, i que el rendiment augmenti, ja que no es perd temps en desenvolupar més tasques de les necessàries o en centrar-se en tasques que no tenen cap valor de negoci.

Centrar el procés de desenvolupament cap al valor de negoci també és molt interessant, ja que implica que el producte resultant complirà, segur, les expectatives del client i dels futurs usuaris, ja que s'han anat a desenvolupar tasques que aquests poden veure, testejar i aplicar; si per

exemple es dedica molt més temps en optimitzar l'estructura del model de dades o qualsevol terme que no és visible als ulls del client, tot i que siguin factors importants a nivell intern, l'usuari no pot veure un resultat real, i per tant no estarà tant satisfet.

A nivell de punts negatius, considero que Lean és més una tècnica, una filosofia, per iniciar un producte que no pas una metodologia de gestió de projectes, ja que no defineix cap mena de timing de producció, ni de creació d'històries d'usuari i per tant, tampoc especifica la creació d'un backlog; només posa l'accent en prioritzar sempre aquells factors amb valor de negoci; per tant, crec que realitzar un projecte de manera organitzada, optimitzada i assegurant un producte testejat i utilitzable, és completament impossible si només s'aplica Lean.

Lean defineix les diferents etapes de vida d'un projecte, passant per alt l'etapa de producció, en la que és completament necessari aplicar una altra metodologia de gestió, ja sigui Kanban, com s'ha fet en aquest projecte, Scrum, eXtreme Programming o qualsevol altra metodologia de gestió de projectes aplicada al desenvolupament de software.

A part, crec que Lean omet la motivació de l'equip de treball, en primera instància, sembla que l'equip és només un conjunt de persones que han de prioritzar les seves tasques cap al benefici econòmic del projecte, i que per aconseguir-ho han de realitzar un aprenentatge constant, però en cap moment busca incentivar o motivar a l'equip, no parla de la potestat de l'equip en la presa de decisions ni de l'organització, jeràrquica o no, d'aquest. Tampoc té en compte la visualització del flux de treball, ni les fases de testeig necessàries per assegurar la qualitat del producte, tampoc té en compte la participació del client o product owner en el procés de desenvolupament.

En conclusió, crec que Lean no es una metodologia de gestió de projectes encarada al desenvolupament de software; crec que és més aviat una metodologia que planteja les diferents fases de producció del producte i es centra molt en el benefici econòmic, el valor de negoci i l'avantatge competitiva del producte, sense centrar-se en la fase de desenvolupament del producte; diguem que dona idees poc definides, com ara l'eliminació del malbaratament de recursos i de temps, i l'orientació al valor de negoci, però sense especificar concretament com fer-ho.

4.3.4 Resultats

Els resultats després d'aplicar Lean en la última fase de desenvolupament de l'aplicació ROAD són els següents:

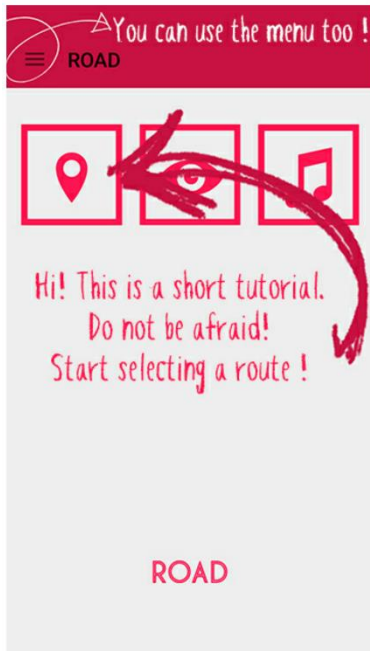


Figura 41. Creació del tutorial al iniciar l'aplicació

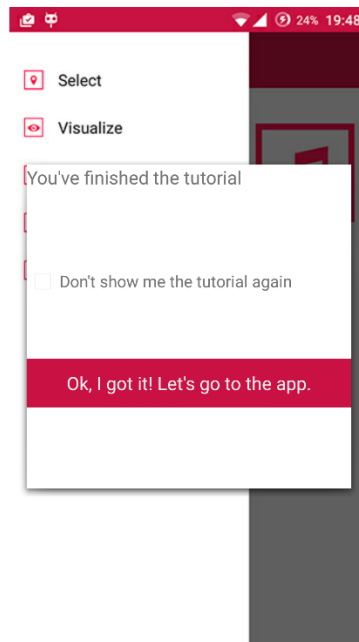


Figura 42. Finalització del tutorial donant l'opció de no tornar-lo a veure

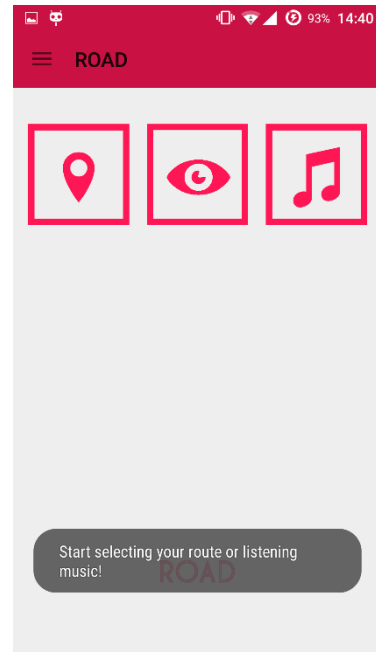


Figura 43. Icones per accedir a les diferents subaplicacions i missatges explicatius

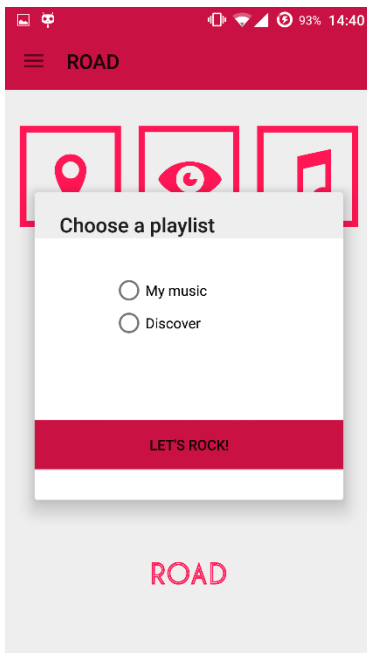


Figura 44. Dialog per seleccionar llista de reproducció



Figura 45. Reproductor per a les cançons del dispositiu

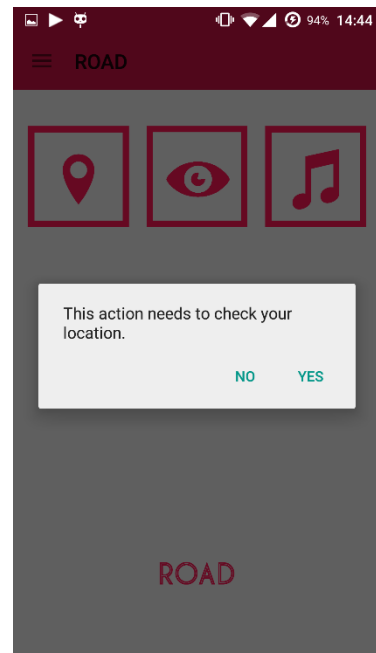


Figura 46. Permisos per activar la localització al escollir l'opció discover

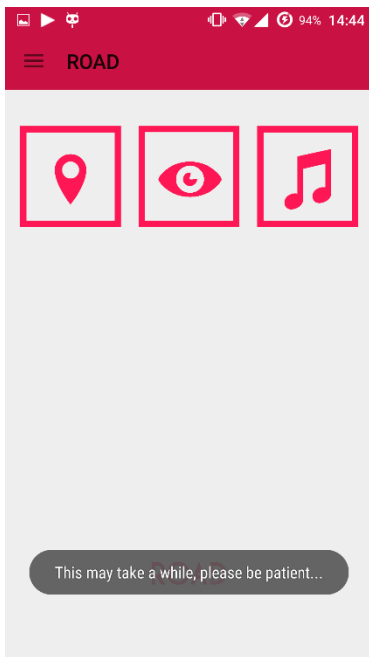


Figura 47. Càrrega de les cançons dels altres usuaris

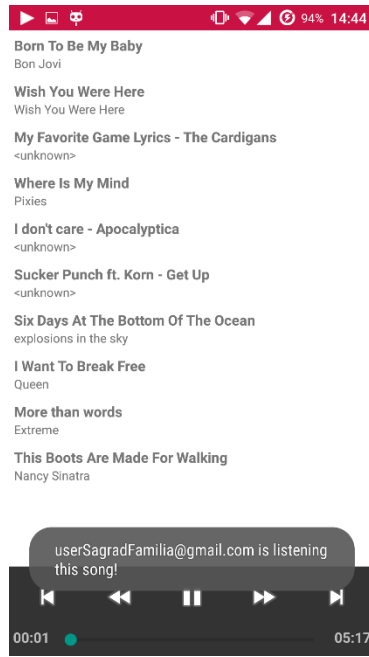


Figura 48. Càrrega de les cançons dels altres usuaris

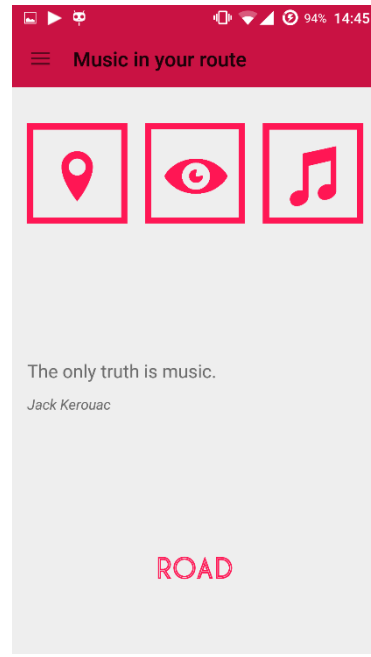


Figura 49. Millores de interfície

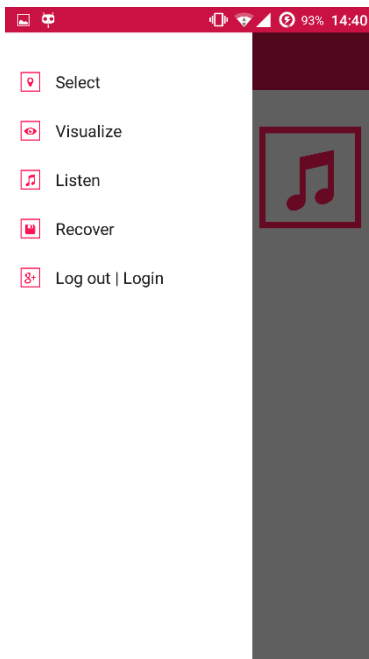


Figura 50. Millores d'interfície

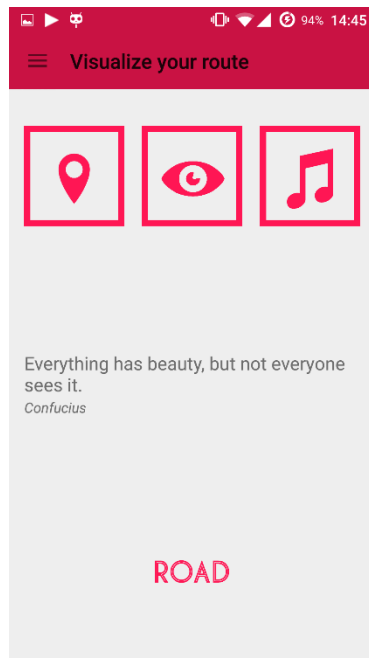


Figura 51. Millores d'interfície

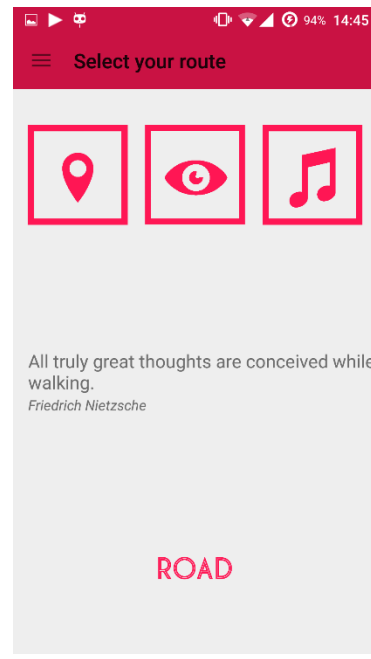


Figura 52. Millores d'interfície

Les figures 40 i 41 fan referència al tutorial mostrat a l'usuari. La figura 42, 50, 51 i 52 ens mostren millores a nivell d'interfície d'usuari i les figures de la 44 a la 49 ens il·lustren la utilització del reproductor de música.

A nivell de codi, es van crear i modificar les següents classes que es poden visualitzar a l'annex número 1 d'aquesta memòria:

- Vista
 - Creació de fragments per al nou menú
 - Inicien les activitats quan es selecciona l'opció dins del menú.
 - MainActivity.java
 - Icones clicables per accedir a les activitats.
 - MusicPlayer.java
 - Activity principal del reproductor de música.
 - Implementa tot el control del Media Player, quan s'escull una cançó, es pausa, s'avança dins de la cançó etc.
 - Inicia un objecte de tipus Music Controller.
 - Inicia un objecte de tipus Music Service.
 - MusicService.java
 - Service que fa funcional el reproductor de música mitjançant la implementació de Media Player.
 - Media Player: s'utilitza per controlar la reproducció d'àudio/vídeo. En el cas de la nostra aplicació s'enllaça directament amb MusicPlayer.java.

Els estats d'una objecte tipus Media Player són els següents:

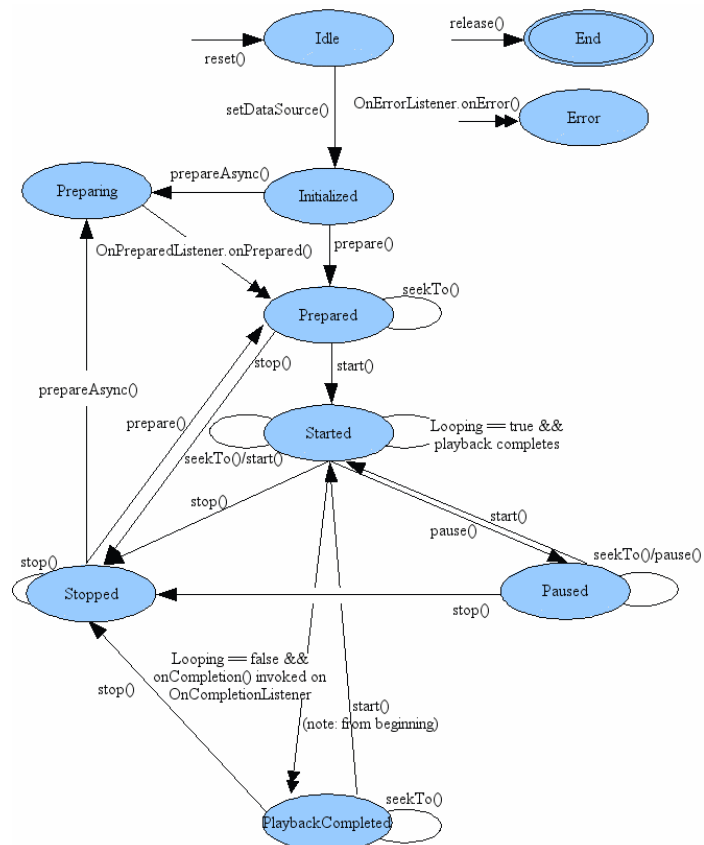


Figura 53. Diagrama d'estats [20].

5

Media Player i les diferents situacions que s'han de donar per passar d'un estat a l'altre.

- Service: un service d'Android és una tasca que s'executa en segon pla sense que l'usuari hagi d'interactuar amb ella. En el cas d'un reproductor de música, es necessita un service perquè un cop l'usuari està escoltant una cançó i surt de l'activity en concret o de l'aplicació, la música ha de continuar sonant.
 - En el nostre service es crea a la barra de notifikacions, una finestra de que ens permet accedir al reproductor de música encara que estiguem fora de l'aplicació.
- TutorialActivity.java
 - Crea un objecte de tipus ImagePagerAdapter que extens de PageAdapter per mostrar les imatges com si fossin diapositives, al fer slide es passa a la següent o anterior imatge.
- LoginActivity.java
 - Modificació del codi per gestionar si es mostra o no el tutorial quan l'usuari ja està loguejat. Si no està loguejat es mostra sempre.

- S'han creat tots els xml corresponents a les activies declarades.
- Controlador
 - MusicController.java
 - Encarregat de mostrar la barra amb els diferents botons i progrés de la cançó al MusicPlayer.java
 - Modificació de funcions per relacionar vista amb model.
- Model
 - GestorPreIniReproductorMusica.java
 - Encarregat d'inicialitzar el Dialog perquè l'usuari esculli la llista de reproducció que vulgui.
 - Crida a la funció per inserir els usuaris amb la seva localització a la cançó a base de dades.
 - Calcula la distància del current user amb els usuaris ficticis de la base de dades.
 - Ordena la llista de reproducció segons distància.
 - Gestiona els canvis de localització del current user utilitzant onLocationChanged.
 - Obtenim la localització de l'usuari cada cop que canvia de posició.
 - InformacionMusica.java
 - Conté la informació necessària que necessiten compartir les activitities
 - El llistat de cançons.
 - Si hi ha una cançó reproduint-se actualment.
 - Si l'usuari està escoltant la seva música o la dels altres usuaris.
 - MySQLLiteHelper.java
 - Creació d'una nova taula anomenada cancion.
 - El nou model de dades és el següent

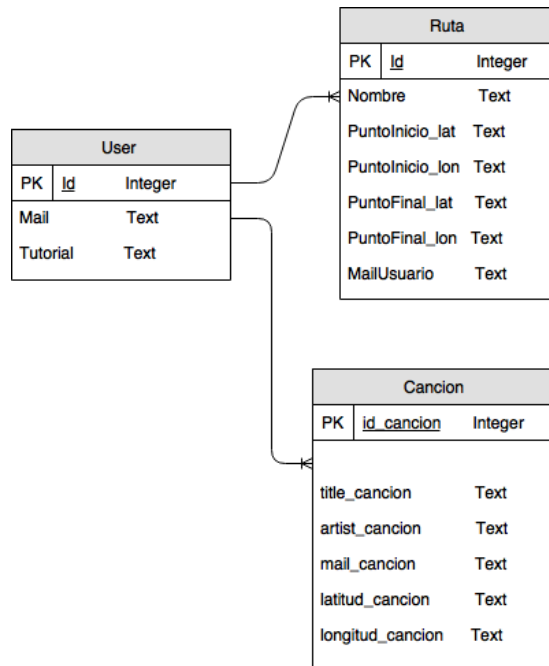


Figura 54. Model de dades.

La figura 54 ens mostra el model de dades on un usuari pot crear n rutes i escoltar n cançons. La nova taula creada en aquest cas és Cancion, que conté una clau primària de tipus Integer, i el títol de la cançó, l'artista, l'e-mail de l'usuari que està escoltant la cançó i el punt on es troba aquest usuari.

- Song.java
 - Conté els atributs necessaris de la cançó.
 - Artista.
 - Mail de l'usuari que l'escolta.
 - Latitud de la posició de l'usuari que l'escolta.
 - Longitud de la posició de l'usuari que l'escolta.
- SoporteReproductorMusica.java
 - Insereix els usuaris ficticis a base de dades.
 - Obté el llistat de cançons.

5. RESULTATS FINALS DE L'APLICACIÓ.

En aquesta secció es mostraran els resultats finals de l'aplicació Road a nivell d'interfície i usabilitat, mostrant com seria l'experiència de l'usuari el primer cop que executa l'aplicació.

A continuació es mostren les captures de pantalla de l'aplicació:

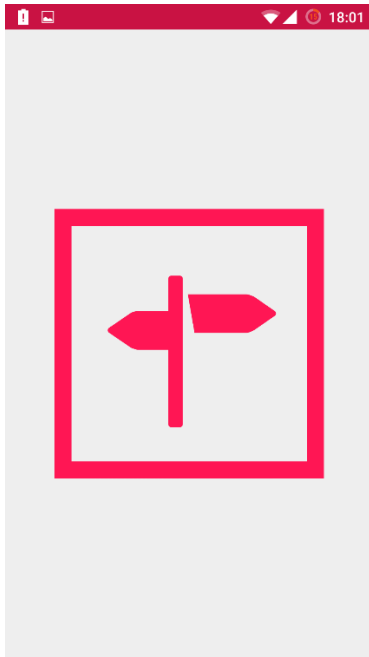


Figura 55. Inici de l'aplicació. Splash Screen.

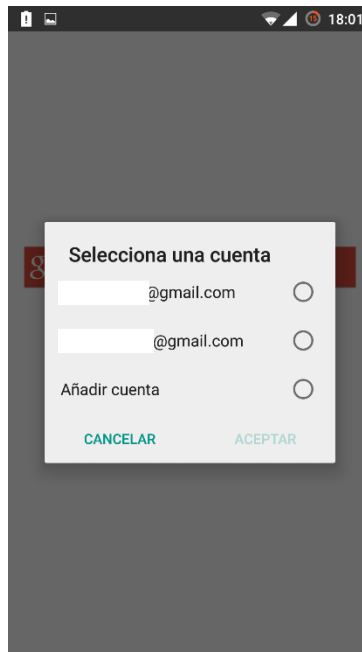


Figura 56. Selecció del compte de Google per a l'inici de sessió.

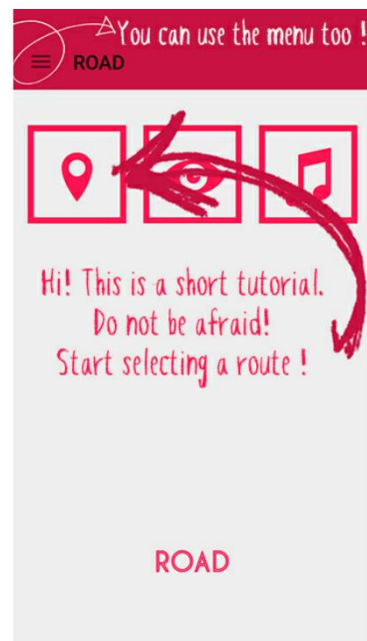


Figura 57. Visualització del tutorial.

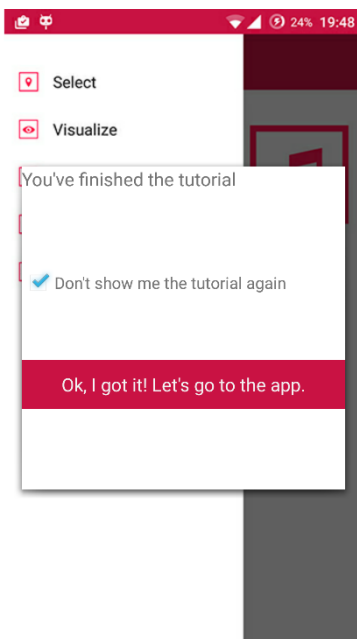


Figura 58. Finalització del tutorial.

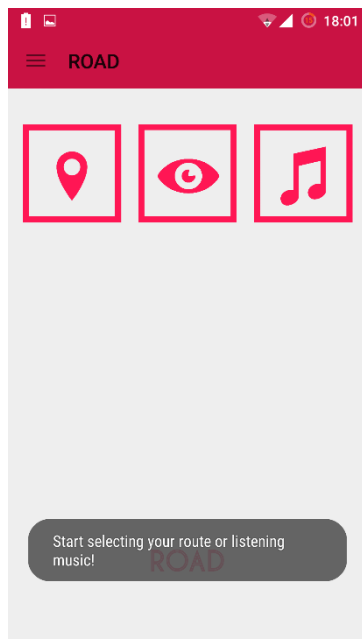


Figura 59. Main Activity. Finestra principal de l'aplicació mitjançant la qual podem accedir a qualsevol subaplicació.

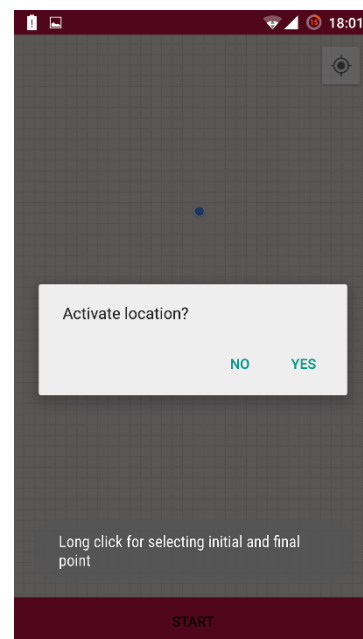


Figura 60. Selecció de la ruta. Permisos de localització.



Figura 61. Localització de l'usuari.

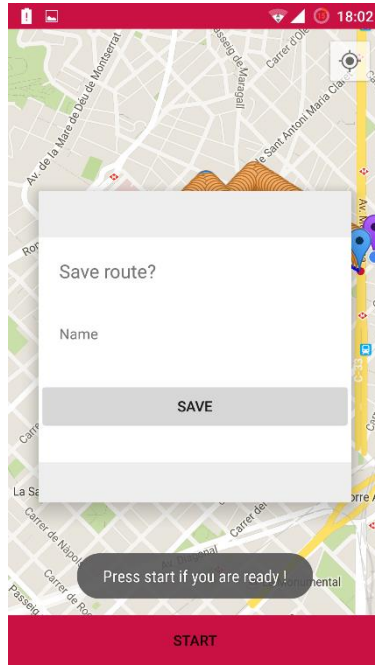


Figura 62. Guardat de la ruta.

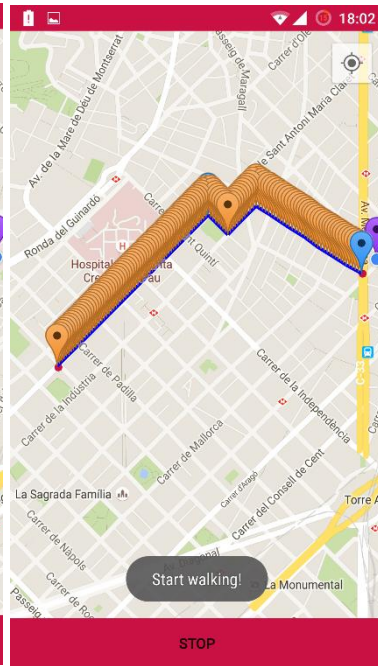


Figura 63. Inici de ruta i visualització del camí.

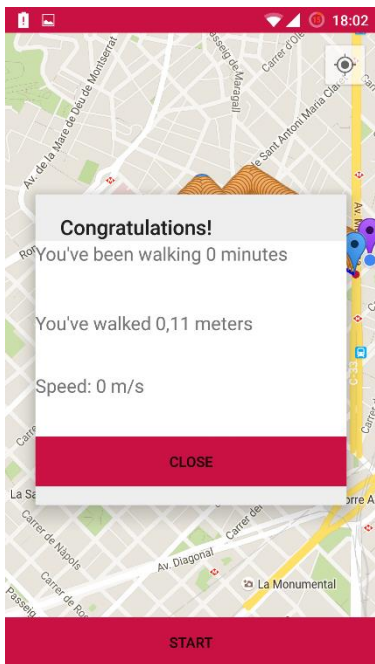


Figura 64. Finalització de la ruta.

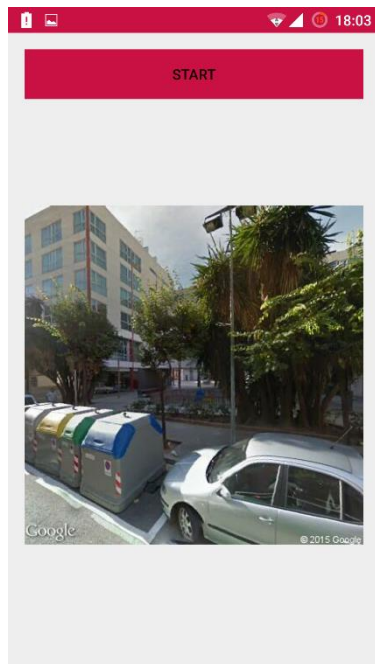


Figura 65. Visualització de la ruta.

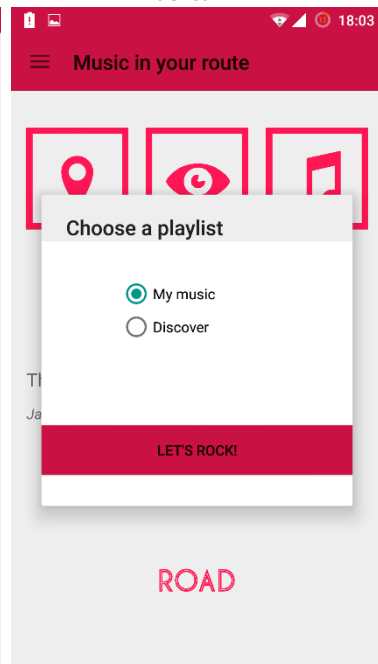


Figura 66. Selecció de la llista de reproducció.



Figura 67. Música del dispositiu.

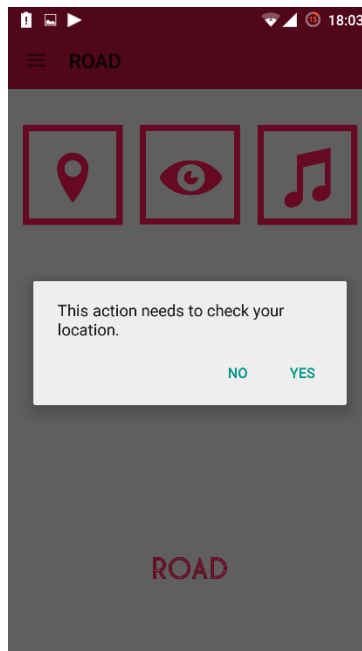


Figura 68. Música dels altres usuaris. Sol·licitud de permisos.

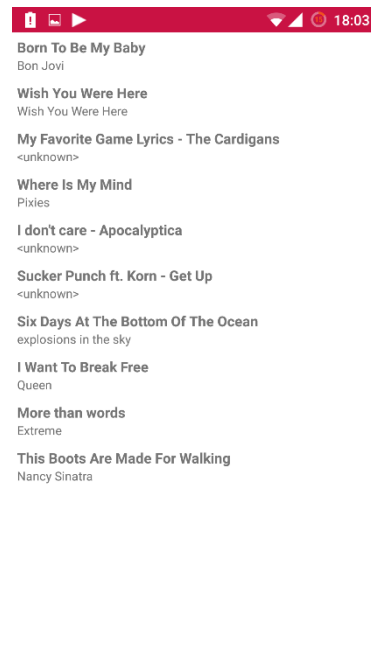


Figura 69. Llista ordenada segons distància dels altres usuaris.

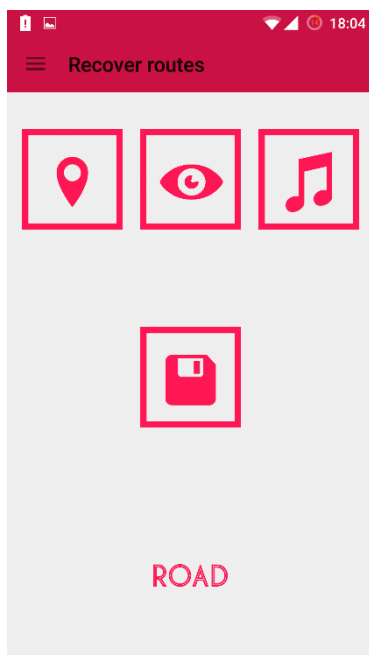


Figura 70. Recuperació de les rutes.

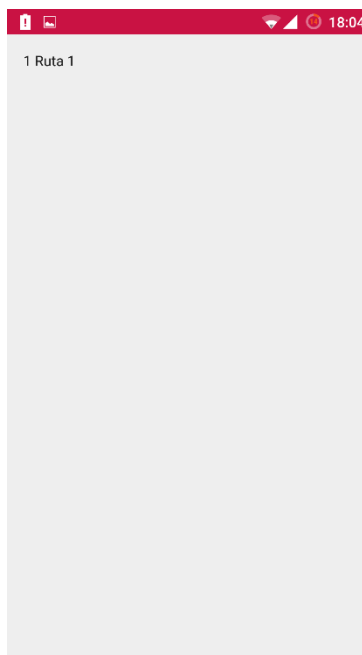


Figura 71. Ruta disponibles per a ser recuperada.

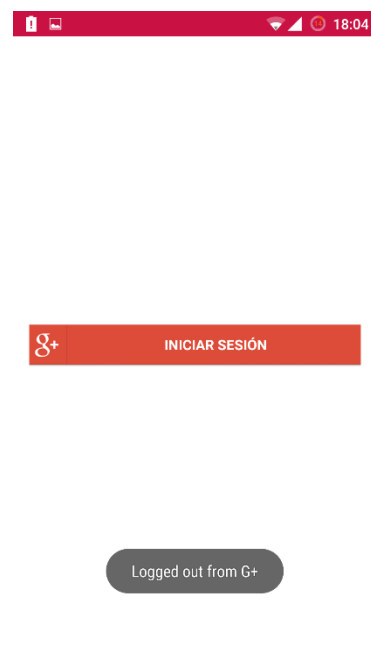


Figura 72. Tancament de sessió de Google.

Com es pot veure mitjançant les figures presentades, es tracta d'una aplicació completament usable, amb una interfície intuïtiva i que compleix tots els requeriments establerts.

El resultat final a nivell de codi es pot veure representat en el diagrama de classes que es pot consultar a l'annex número 1 de la present memòria.

Les figures de la 55 a la 58 ens mostren l'inici de l'aplicació, amb la pantalla de presentació, el registre i la visualització del tutorial. Les figures de la 59 a la 64 fan referència a la selecció de la ruta i el seguiment que ofereix l'aplicació un cop es finalitza.

Seguidament, a la figura 65 trobem l'exemplificació de la visualització de la ruta, per, de la 66 a la 69, veure el funcionament del reproductor de música.

Finalment, les figures 70 i 71 fan referència a la recuperació de rutes, i la última, la número 72, ens mostra el tancament de sessió de Google.

Per tal de testejar correctament el funcionament de l'aplicació es van realitzar múltiples proves durant tot el procés de desenvolupament, incloent a usuaris externs perquè provessin l'aplicació i així fer millores a nivell d'usabilitat, com ara la creació del tutorial i els missatges explicatius perquè l'experiència d'usuari fos més fàcil. A part, el fet d'involucrar usuaris externs va permetre detectar errors que, generalment, la persona que ha realitzat l'aplicació no pot detectar ja que sap com fer-la funcionar correctament.

Un cop finalitzat el desenvolupament i les fases de testeig s'ha obtingut una aplicació sòlida amb una correcta gestió d'errors i que ofereix una experiència d'usuari simplificada i intuïtiva.

6. CONCLUSIONS FINALS I COMPARATIVA.

Un cop finalitzat l'estudi de les metodologies de gestió de projectes àgils Scrum, Kanban i Lean, i acabat el desenvolupament de l'aplicació Android anomenada ROAD, podem concloure que s'han complert correctament tots els objectius establerts.

Primerament, s'ha pogut realitzar un profund anàlisi teòric de tres de les metodologies de gestió de projectes més populars avui en dia; hi ha hagut una fase de documentació i estudi de les diferents metodologies per tal de poder-les aplicar correctament en el context d'aquest projecte i de l'aplicació.

Ha sigut necessari entendre els antecedents històrics de cada metodologia per entendre la seva filosofia, i també ha sigut necessari documentar-se sobre l'antítesi de les metodologies àgils, per aplicar correctament el cicle de vida iteratiu i incremental i no caure en una estructuració de projectes amb un cicle de vida predictiu.

Un cop realitzat l'estudi teòric de les metodologies, s'han identificat les seves característiques i s'han posat en pràctica, dividint l'aplicació Android en tres fases de desenvolupament, assignant a cada fase una metodologia de gestió diferent.

Relatiu a Scrum, s'han pogut aplicar la totalitat de les característiques tenint en compte que el projecte s'ha realitzat a nivell individual; el backlog prioritzat amb històries d'usuari, la divisió del treball en sprints, els daily meetings, el tancament d'sprint i l'sprint review, la realització de les taules amb les estimacions i el recompte d'hores i finalment la creació dels Burndown Charts per a l'anàlisi del rendiment.

A nivell de l'aplicació de Kanban, també s'han pogut realitzar totes les característiques, s'ha visualitzat el flux de treball en una pissarra dividida en columnes, representant les fases del procés de desenvolupament, i amb targetes representant les tasques realitzades. S'ha limitat el WIP en les columnes necessàries i s'ha aplicat la metodologia de *pull*, redactant polítiques d'acceptació per passar d'una fase a una altra de desenvolupament.

Finalment, relatiu a Lean, s'ha pogut aplicar relativament, només s'ha seleccionat d'aquesta metodologia l'eliminació dels Seven Wastes of Lean, enfocant el desenvolupament a la reducció de malbaratament a nivell de codi, temps i rendiment. S'han prioritzat les tasques segons el seu valor de negoci i benefici per l'usuari final i el product owner. Per visualitzar el flux de treball i mantenir un seguiment del procés de desenvolupament ha sigut necessari utilitzar la metodologia Kanban.

Tot i que una de les principals característiques d'aquestes metodologies és un equip multidisciplinari i auto-gestionat, ha sigut completament viable aplicar totes les característiques en un projecte individual, sent totes una gran eina d'organització del procés de desenvolupament i una gran ajuda en l'optimització de recursos per millorar el rendiment i l'entrega de software acabat i funcional.

S'ha pogut realitzar un estudi teòric-aplicat de diferents metodologies de gestió de projectes àgils, implicant un profund estudi teòric de la filosofia Agile per a la gestió de projectes i en concret d'Scrum, Kanban i Lean, incloent la comprensió de la situació actual d'aquestes metodologies partint dels seus antecedents històrics i de l'anàlisi de les seves principals característiques, actors involucrats, cerimònies realitzades i artefactes generats.

Paral·lelament, s'ha realitzat una aplicació Android que ha complert totes les històries d'usuari plantejades i per tant, els objectius i característiques necessàries per a ser completament funcional. El desenvolupament de l'aplicació ha hagut de complir estrictes polítiques d'acceptació i testeig per a considerar-se acabada.

En conclusió, s'han complerts tot els objectius i un cop acabat el projecte s'ha obtingut un coneixement sòlid i profund de les metodologies estudiades, així com de la filosofia Agile, podent aplicar aquestes metodologies al desenvolupament de l'aplicació ROAD, que ha complert tots els requeriments d'usuari.

7. REFERÈNCIES I BIBLIOGRAFIA.

Agile:

1. Jummpsblog, "*Agile*", 2014, <https://jummp.wordpress.com>
2. J.Garzas, "*Veterano ciclo de vida iterativo e incremental*", <http://www.javiergarzas.com/2010/01/veterano-ciclo-de-vida-iterativo-incremental.html>
3. Jummpsblog, "*Desarrollo de software. Ciclo de vida iterativo incremental*", 2011, <https://jummp.wordpress.com>
4. APMG International. Agile PM, "*¿Qué es Agile Project Management?*", <http://www.qrpinternational.es>
5. M.Goikolea, "*¿Qué es Agile Project Management? Ventajas de ser el más rápido y ágil*", 2014, <http://comunidad.iebschool.com>
6. D.Buonamico, "*Historia de las metodologías ágiles en contexto*", 2013, <http://www.caminoagil.com>

Scrum:

7. H.Takeuchi i I.Nonaka, "*The new product development game*", 1986, <https://hbr.org/1986/01/the-new-new-product-development-game>
8. K.S.Rubin, "*Metodologías Ágiles: Orígenes de Scrum*", 2013, <http://www.gary.pe>

Kanban:

9. J.López, "*Mejora tu Trabajo en equipo con el método Kanban*", 2013, <http://hipertextual.com>
10. Scrum Manager, "*Kanban: origen y definición*", 2013, <http://www.scrummanager.net>
11. W.Sterlin, "*Origen de Kanban y nacimiento del sistema pull*", 2011, <http://qtclean.forosactivos.net>
12. J.Boeg, "*Priming Kanban*", 2011, <http://www.infoq.com/minibooks/priming-kanban-jesper-boeg>

Lean:

13. V.K.Bandaru, *"How to Manage the "7 Wastes" of Agile Software Development"*, 2013, <https://www.scrumalliance.org>
14. Instituto Andaluz del Lean, *"Historia de la metodologia Lean"*, <http://ialean.org>
15. Lean College. Grupo MDC, *"Historia"*, <http://www.leanmanufacturing.org>
16. C.Larman i V.R.Basili, *"Iterative and Incremental Development: A Brief History"*, 2003, <http://www.craiglarman.com/wiki/downloads/misc/history-of-iterative-larman-and-basili-ieee-computer.pdf>
17. Agile Alliance, *"Invest"*, 2013, <http://guide.agilealliance.org/guide/invest.html>

Android:

18. J.Cancela, *"Programando en Android – Conceptos Iniciales"*, 2008, <http://javiercancela.com>
19. Android Developers, *"Activity"*, 2015, <http://developer.android.com>
20. Android Developers, *"Service"*, 2015, <http://developer.android.com>

ANNEX NÚMERO 1

