



PIANO DI QUALIFICA

Gruppo Cyber13 - Progetto P2PCS

Informazioni sul documento

Versione	1.0.0
Data Redazione	04/04/2019
Responsabili	Elena Pontecchiani Andrea Casagrande
Redazione	Daniel Mirel Bira Andrea Casagrande Ilaria Rizzo Matteo Squeri
Verifica	Fabio Garavello Ilaria Rizzo
Approvazione	Andrea Casagrande
Uso	Esterno
Destinatari	GaiaGo Cyber13 Prof. Tullio Vardanega Prof. Riccardo Cardin
Mail di contatto	swe.cyber13@gmail.com

Diario delle modifiche

Versione	Data	Descrizione	Autore	Ruolo
1.0.0	04/04/2019	Approvazione per rilascio RR conforme alle norme	Andrea Casagrande	Responsabile
0.3.0	04/04/2019	Verifica generale con esito positivo per il rilascio del documento in RR e aggiunta appendice di riepilogo	Ilaria Rizzo	Verificatore
0.2.0	25/03/2019	Verifica con esito positivo dei contenuti della sezione 1 e 3	Fabio Garavello	Verificatore
0.1.0	25/03/2019	Verifica con esito positivo dei contenuti della sezione 2	Ilaria Rizzo	Verificatore
0.0.5	22/03/2019	Stesura della sezione 3.3.3 sulla gamification	Andrea Casagrande	Analista
0.04	16/03/2019	Stesura della sezione 3	Ilaria Rizzo, Matteo Squeri	Analisti

0.03	14/03/2019	Stesura della sezione 2	Daniel Mirel Bira	Analista
0.0.2	12/03/2019	Stesura sezione di introduzione (sezione 1)	Ilaria Rizzo	Analista
0.0.1	12/03/2019	Creazione scheletro documento	Ilaria Rizzo	Analista

Indice

1	Introduzione	7
1.1	Scopo del documento	7
1.2	Scopo del prodotto	7
1.3	Glossario	7
1.4	Riferimenti	7
1.4.1	Riferimenti normativi	7
1.4.2	Riferimenti informativi	7
2	Qualità di processo	9
2.1	PROC[0001] - Project assessment and Control Process	10
2.1.1	Obiettivi	11
2.1.2	Strategie	11
2.1.3	Metriche	11
2.1.3.1	M[PROC][0001] - Schedule Variance	11
2.1.3.2	M[PROC][0002] - Budget Variance	11
2.2	PROC[0002] - Risk Management Process	11
2.2.1	Obiettivi	11
2.2.2	Strategie	11
2.2.3	Metriche	12
2.2.3.1	M[PROC][0003] - Rischi non individuati	12
2.3	PROC[0003] - Software Detailed Design Process	12
2.3.1	Obiettivi	12
2.3.2	Strategie	12
2.3.3	Metriche	12
2.3.3.1	M[PROC][0004]: Numero campi dati per classe	12
2.3.3.2	M[PROC][0005] - Metodi per classe	12
2.3.3.3	M[PROC][0006] - Parametri per metodo	12
2.3.3.4	M[PROC][0007]: Grado di instabilità	13
2.4	PROC[0004] - Software Construction Process	13
2.4.1	Obiettivi	13
2.4.2	Strategie	13
2.4.3	Metriche	13
2.4.3.1	M[PROC][0008] - Complessità Ciclomatica	13
2.4.3.2	M[PROC][0009] - Linee di codice per linee di comando	13
2.4.3.3	M[PROC][0010] - Halstead Difficulty per-function	13
2.4.3.4	M[PROC][0011] - Halstead Volume per-function	14
2.4.3.5	M[PROC][0012] - Halstead Effort per-function	14
2.4.3.6	M[PROC][0013] - Indice di manutenibilità	14
2.5	PROC[0005] - Test	14
2.5.1	Scopo	14
2.5.1.1	M[PROC][TM][0001] - Percentuale di codice coperto da test	14
2.5.1.2	M[PROC][TM][0002] - Percentuale di test passati	14
2.5.1.3	M[PROC][TM][0003] - Percentuale di test non passati	14

3	Qualità di prodotto	15
3.1	Scopo	15
3.2	Qualità dei documenti	15
3.2.1	Obiettivi: Comprensione	15
3.2.2	Metriche per documenti	15
3.2.2.1	M[PROD][D][0001]: Indice di <i>Gulpease_g</i>	15
3.2.2.2	M[PROD][D][0002]: Errori ortografici	15
3.3	Qualità del software	15
3.3.1	Obiettivi	15
3.3.1.1	Funzionalità	15
3.3.1.1.1	M[PROD][S][0001]: Copertura requisiti obbligatori	16
3.3.1.1.2	M[PROD][S][0002]: Copertura requisiti accettati	16
3.3.1.1.3	M[PROD][S][0003]: Accuratezza rispetto alle attese	16
3.3.1.2	Affidabilità	16
3.3.1.2.1	M[PROD][S][0004]: Densità di <i>failure_g</i>	16
3.3.1.2.2	M[PROD][S][0005]: Blocco di operazioni non corrette	16
3.3.1.3	Efficienza	16
3.3.1.3.1	M[PROD][S][0006]: Tempo di risposta	16
3.3.1.4	Usabilità	17
3.3.1.4.1	M[PROD][S][0007]: Comprensibilità delle funzioni offerte	17
3.3.1.4.2	M[PROD][S][0008]: Facilità di apprendimento delle funzionalità	17
3.3.1.4.3	M[PROD][S][0009]: Utilizzo effettivo delle funzionalità	17
3.3.1.5	Manutenibilità	17
3.3.1.5.1	M[PROD][S][0010]: Capacità di analisi di failure	17
3.3.1.5.2	M[PROD][S][0011]: Impatto delle modifiche	17
3.3.1.6	M[PROD][S][0012]: Rapporto linee di commento su linee di codice	18
3.3.2	M[PROD][S][0013]: Versioni di Android supportate	18
3.3.3	Gamification e framework Octalysis	18
3.3.3.1	Cos'è la gamification	18
3.3.3.2	Il framework Octalysis	18
3.3.3.3	Gamification applicata al progetto P2PCS	19
3.3.3.4	M[PROD][S][0014]: Copertura del framework Octalysis	21
A	Resoconto attività di verifica	22
A.1	Revisione dei Requisiti	22
A.1.1	Metriche processi	22

A.1.2	Metriche prodotto	23
A.1.3	M[PROD][D][0001]: Indice di Gulpease	23
A.1.4	M[PROD][S][0014]: Copertura del framework Octalysis	24

Elenco delle figure

1	Grafico valori indice Gulpease	23
2	Grafico copertura Core Drives Octalysys	25

1 Introduzione

1.1 Scopo del documento

Lo scopo di tale documento consiste nello stabilire le norme utilizzate per verificare la qualità del prodotto e del processo. A tale scopo sarà svolta una continua attività di verifica per rilevare e correggere eventuali anomalie tramite metriche di seguito descritte.

1.2 Scopo del prodotto

Lo scopo del prodotto è quella di realizzare un'applicazione *Android_{|g|}* che implementi un servizio di car sharing *peer-to-peer_{|g|}*.

1.3 Glossario

Onde evitare ambiguità o incomprensioni di natura lessicale, si allega il *Glossario v1.0.0*. All'interno del documento saranno presenti parole di ambito specifico, uso raro che potrebbero creare incomprensioni. Per una maggiore leggibilità tali parole sono riconoscibili all'interno dei vari documenti in quanto scritte in corsivo e con un 'g' a pedice tra barre orizzontali (per esempio *Glossario_{|g|}*)

1.4 Riferimenti

1.4.1 Riferimenti normativi

- *Norme di Progetto v1.0.0*
- Capitolato C5: Car sharing peer to peer.
<https://www.math.unipd.it/~tullio/IS-1/2018/Progetto/C5.pdf>

1.4.2 Riferimenti informativi

- Slide corso Ingegneria del Software:
<https://www.math.unipd.it/~tullio/IS-1/2018/Dispense/L03.pdf>
- Slide corso Ingegneria del Software: Qualità di prodotto
<https://www.math.unipd.it/~tullio/IS-1/2018/Dispense/L13.pdf>
- Slide corso Ingegneria del Software: Qualità di processo
<https://www.math.unipd.it/~tullio/IS-1/2018/Dispense/L14.pdf>
- Standard ISO/IEC 9126
https://it.wikipedia.org/wiki/ISO/IEC_9126

- Indice Gulpease:
https://it.wikipedia.org/wiki/Indice_Gulpease
- Framework Octalysis:
<https://yukaichou.com/gamification-examples/>
- ISO/IEC 15504:
https://en.wikipedia.org/wiki/ISO/IEC_15504

2 Qualità di processo

La qualità finale di un prodotto è determinata in maniera decisiva dalla qualità dei processi che portano alla produzione dello stesso, per questo motivo il team ha deciso di adottare lo standard ISO/IEC 15504 conosciuto anche come *SPICE*_{|g|} il quale definisce un modello di valutazione dello stadio di maturità di un processo. SPICE prevede sei livelli di maturità per un dato processo e per ognuno di essi definisce anche degli attributi di processo che permettono di determinare con maggior precisione se un livello di maturità viene raggiunto o meno dal processo in questione. In questo documento verranno citate solo le voci ritenute rilevanti in relazione al contesto.

Level 0 - Incomplete process: il processo non è implementato o non raggiunge gli obiettivi prefissati, gli output del processo sono pochi o inesistenti. Gli attributi di tale processo sono:

- non sono previsti attributi di processo per questo livello.

Level 1 - Performed process: il processo viene attuato e raggiunge gli obiettivi prefissati, tuttavia non viene scrupolosamente controllato. Gli attributi di tale processo sono:

- **P.A. 1.1 - Performed process:** capacità di raggiungere i propri obiettivi e produrre output identificabili.

Level 2 - Managed process: il processo viene attuato, controllato, tracciato e gli output prodotti raggiungono standard prefissati. Gli attributi di tale processo sono:

- **P.A. 2.1 - Performance management:** capacità di produrre output che raggiungono gli obiettivi prefissati;
- **P.A. 2.2 - Work product management:** capacità di produrre output controllato e tracciato.

Level 3 - Established process: il processo viene attuato e controllato seguendo i principi dell'ingegneria del software. Gli attributi di tale processo sono:

- **P.A. 3.1 - Process definition:** capacità di produrre output che si attengano agli standard dell'ingegneria del software;
- **P.A. 3.2 - Process resource:** capacità di produrre output efficacemente utilizzando una quantità di risorse ragionevole.

Level 4 - Predictable process: il processo viene attuato con vincoli determinati a raggiungere gli obiettivi previsti. Il processo risulta essere ben collaudato nella pratica. Gli attributi di tale processo sono:

- **P.A. 4.1 - Process measurement:** capacità di utilizzare le misure ottenute durante l'esecuzione del processo per verificare in futuro il raggiungimento degli obiettivi prefissati;

- **P.A. 4.2 Process control:** capacità di modificare l'esecuzione del processo in seguito ai dati raccolti

Level 5 - Optimizing process: il processo ha una certa consistenza nel raggiungere i propri obiettivi, viene ottimizzato per adempiere al meglio agli obiettivi correnti e futuri. gli attributi di tale processo sono:

- **P.A. 5.1 - Process change:** capacità di tracciare tutti i cambiamenti del processo, siano essi strutturati o di esecuzione;
- **P.A. 5.2 - Continuous improvement:** capacità di implementare le modifiche applicate.

Per tutti gli attributi di processo, SPICE fornisce un metodo di valutazione per misurare il loro raggiungimento:

- **N:** non posseduto (0% - 15%);
- **P:** parzialmente posseduto (16% - 50%);
- **L:** largamente posseduto (51% - 85%);
- **F:** totalmente posseduto (86% - 100%).

Tali valori possono essere utilizzati nel ciclo $PDCA_{|g|}$ il cui scopo è quello di controllare la qualità di un processo durante tutto il suo ciclo di vita e permettere il miglioramento in efficacia ed efficienza dello stesso. Le fasi descritte da PDCA sono le seguenti:

- **Plan:** fase di pianificazione dove si decidono e si individuano gli obiettivi di qualità e i risultati desiderati;
- **Do:** fase in cui si mette in atto il piano stabilito nella fase precedente;
- **Check:** fase di verifica in cui si confrontano i dati in output della fase Do con i risultati previsti in fase Plan;
- **Act:** fase in cui si individuano le cause delle eventuali discordanze riscontrate in fase di Check e si determinano le azioni da intraprendere per risolvere tali discordanze e migliorare il processo aumentandone così la qualità.

Il gruppo ha inoltre individuato nello standard ISO/IEC 12207 alcuni processi il cui scopo è quello di garantire la qualità del prodotto finale. In questo documento verranno citate solo le voci ritenute rilevanti in relazione al contesto.

2.1 PROC[0001] - Project assessment and Control Process

Lo scopo di questo processo è quello di determinare lo stato del lavoro svolto ed assicurare che il tutto si stia svolgendo secondo i piani ed entro i limiti di risorse e tempo prestabiliti.

2.1.1 Obiettivi

- Ogni membro del gruppo svolgerà il $task_{|g|}$ assegnatogli nei tempi previsti;
- Le risorse impiegate per una fase non dovranno superare i limiti prestabiliti.

2.1.2 Strategie

Il *Project Manager*_{|g|} deve monitorare lo svolgimento del processo in modo da rilevare il prima possibile eventuali ritardi nello svolgimento dei task e/o un utilizzo delle risorse superiore ai limiti prestabiliti. In caso di rilevamento di tale eccedenza, il gruppo dovrà assolutamente risolvere il problema entro la data prevista per la consegna finale del prodotto.

2.1.3 Metriche

2.1.3.1 M[PROC][0001] - Schedule Variance

- Range di accettazione: ≥ 0
- Range ottimale: ≥ 0

2.1.3.2 M[PROC][0002] - Budget Variance

- Range di accettazione: ≥ 0
- Range ottimale: ≥ 0

2.2 PROC[0002] - Risk Management Process

Lo scopo di questo processo è quello di individuare, analizzare e monitorare i rischi durante l'intera durata del progetto.

2.2.1 Obiettivi

- Il gruppo individuerà i rischi nella prima fase del progetto e ne terrà traccia fino a quando il rischio non sarà più una possibile evenienza;
- L'individuazione dei rischi verrà svolta ad ogni fase in modo da identificare nuovi possibili rischi introdotti dalle attività svolte nella fase precedente.

2.2.2 Strategie

Il gruppo dovrà tenere sempre sotto stretta osservazione tutti i rischi in modo da poter mitigare al meglio l'eventuale manifestazione.

2.2.3 Metriche

2.2.3.1 M[PROC][0003] - Rischi non individuati

- **Range di accettazione:** 0 - 4
- **Range ottimale:** 0

2.3 PROC[0003] - Software Detailed Design Process

Lo scopo di questo processo è fornire una progettazione di dettaglio del prodotto che andrà ad implementare i requisiti individuati.

2.3.1 Obiettivi

- Il grado di dettaglio della progettazione deve fornire sufficiente informazione per procedere alla codifica e testing di un'unità senza bisogno di ulteriori informazioni.

2.3.2 Strategie

Le componenti individuate durante l'analisi verranno suddivise in piccole unità codificabili e testabili facilmente.

2.3.3 Metriche

2.3.3.1 M[PROC][0004]: Numero campi dati per classe

- **Range di accettazione:** 0-16
- **Range ottimale:** 0-10

2.3.3.2 M[PROC][0005] - Metodi per classe

- **Range di accettazione:** 1-10
- **Range ottimale:** 1-7

2.3.3.3 M[PROC][0006] - Parametri per metodo

- **Range di accettazione:** 0-8
- **Range ottimale:** 0-5

2.3.3.4 M[PROC][0007]: Grado di instabilità

- Range di accettazione: 0-0.8
- Range ottimale: 0.3-0.7

2.4 PROC[0004] - Software Construction Process

Lo scopo di questo processo è definire le attività principali volte alla produzione di unità software.

2.4.1 Obiettivi

- il codice prodotto dovrà risultare di bassa complessità per facilitarne la verifica e la complessità;
- sdoppiamenti di flusso verranno ridotti al minimo necessario;
- il codice prodotto dovrà risultare facilmente manutenibile.

2.4.2 Strategie

Il team si impegna a mantenere una complessità bassa nella stesura del codice.

2.4.3 Metriche

2.4.3.1 M[PROC][0008] - Complessità Ciclomatica

- Range di accettazione: 1-15
- Range ottimale: 1-10

2.4.3.2 M[PROC][0009] - Linee di codice per linee di comando

- Range di accettazione: ≥ 25
- Range ottimale: ≥ 30

2.4.3.3 M[PROC][0010] - Halstead Difficulty per-function

- Range di accettazione: 0-25
- Range ottimale: 0-10

2.4.3.4 M[PROC][0011] - Halstead Volume per-function

- Range di accettazione: 20-1500
- Range ottimale: 20-1000

2.4.3.5 M[PROC][0012] - Halstead Effort per-function Rappresenta il costo necessario a scrivere il codice di una funzione.

- Range di accettazione: 0-400
- Range ottimale: 0-300

2.4.3.6 M[PROC][0013] - Indice di manutenibilità

- Range di accettazione: 100-171
- Range ottimale: 120-171

2.5 PROC[0005] - Test**2.5.1 Scopo**

Lo scopo è di definire una misurazione per l'esecuzione dei test e relativi fallimenti

2.5.1.1 M[PROC][TM][0001] - Percentuale di codice coperto da test

- Range di accettazione: $\geq 0\%$
- Range ottimale: 100%

2.5.1.2 M[PROC][TM][0002] - Percentuale di test passati

- Range di accettazione: 100%
- Range ottimale: 100%

2.5.1.3 M[PROC][TM][0003] - Percentuale di test non passati

- Range di accettazione: 0%;
- Range ottimale: 0%.

3 Qualità di prodotto

3.1 Scopo

A seguito dello studio dello standard ISO/IEC 9126 il gruppo ha deciso di adottarne alcuni concetti per garantire la qualità del prodotto, esaminata di seguito con obiettivi e metriche corrispondenti.

3.2 Qualità dei documenti

I documenti devono essere corretti dal punto di vista ortografico, sintattico, logico, e a livello di leggibilità e comprensibilità.

3.2.1 Obiettivi: Comprensione

Il documento deve presentarsi corretto dal punto di vista ortografico e con una leggibilità almeno per persone con licenza media.

3.2.2 Metriche per documenti

3.2.2.1 M[PROD][D][0001]: Indice di *Gulpease*_{|g|}

- Range di accettazione: 50-100;
- Range ottimale: 60-100.

3.2.2.2 M[PROD][D][0002]: Errori ortografici

- Range di accettazione: 100% corretto;
- Range ottimale: 100% corretto.

3.3 Qualità del software

3.3.1 Obiettivi

Lo standard ISO/IEC 9126 scelto dal gruppo stabilisce che il modello di qualità sia rappresentato dalle seguenti sei caratteristiche generali, e dalle loro sotto categorie:

3.3.1.1 Funzionalità

Il gruppo si impegna a creare un prodotto software in grado di soddisfare al 100% i requisiti obbligatori previsti in *Analisi Dei Requisiti v1.0.0*(Accuratezza) fornendo un appropriato insieme di funzionalità (Appropriatezza).

3.3.1.1.1 M[PROD][S][0001]: Copertura requisiti obbligatori

- Range di accettazione: 100%
- Range ottimale: 100%

3.3.1.1.2 M[PROD][S][0002]: Copertura requisiti accettati

- Range di accettazione: 0-100%
- Range ottimale: 100%

3.3.1.1.3 M[PROD][S][0003]: Accuratezza rispetto alle attese

- Range di accettazione: 75-100%
- Range ottimale: 100%

3.3.1.2 Affidabilità

Allo scopo di creare un prodotto software affidabile verranno presi in considerazione i seguenti parametri. Maturità, intesa come capacità dell'applicativo di evitare più possibile errori e malfunzionamenti. Tolleranza agli errori, intesa come capacità invece di gestire tali errori e malfunzionamenti qualora si verificassero.

3.3.1.2.1 M[PROD][S][0004]: Densità di *failure*_{|g|}

- Range di accettazione: 0-10%
- Range ottimale: 0%

3.3.1.2.2 M[PROD][S][0005]: Blocco di operazioni non corrette

- Range di accettazione: 80-100%
- Range ottimale: 100%

3.3.1.3 Efficienza

Il software deve essere nel miglior modo prestante includendo l'utilizzo di quantità e tipo di risorse in maniera adeguata (Utilizzo delle risorse); inoltre deve fornire adeguati tempi di risposta (Comportamento rispetto al tempo);

3.3.1.3.1 M[PROD][S][0006]: Tempo di risposta

- Range di accettazione: 0-8 sec;
- Range ottimale: 0-3 sec.

3.3.1.4 Usabilità

Il prodotto deve essere facilmente comprensibile dall'utente e allo stesso tempo soddisfare le sue esigenze. Nello specifico deve essere chiaro lo scopo di ogni funzionalità e il modo per utilizzarla (Comprensibilità); l'apprendimento dell'uso dell'app deve essere facile e intuitivo (Apprendibilità); le funzioni presenti devono soddisfare le aspettative dell'utente (Operabilità) ed infine l'applicazione deve essere interessante per l'utente (Attrattiva).

3.3.1.4.1 M[PROD][S][0007]: Comprensibilità delle funzioni offerte

- **Range di accettazione:** 80-100%;
- **Range ottimale:** 90-100%.

3.3.1.4.2 M[PROD][S][0008]: Facilità di apprendimento delle funzionalità

- **Range di accettazione:** 0-20 min;
- **Range ottimale:** 0-10 min.

3.3.1.4.3 M[PROD][S][0009]: Utilizzo effettivo delle funzionalità

- **Range di accettazione:** 50-100%;
- **Range ottimale:** 60-100%.

3.3.1.5 Manutenibilità

È la capacità del codice di essere modificato per correzioni e miglioramenti futuri. Per questo motivo il software deve presentare una facilità di analisi per trovare l'errore (Analizzabilità), la possibilità di implementare una modifica (Modificabilità), la capacità di essere facilmente testato (Testabilità) e di non provocare errori dopo la modifica (Stabilità).

3.3.1.5.1 M[PROD][S][0010]: Capacità di analisi di failure

- **Range di accettazione:** 60-100%;
- **Range ottimale:** 80-100%.

3.3.1.5.2 M[PROD][S][0011]: Impatto delle modifiche

- **Range di accettazione:** 0-20%;
- **Range ottimale:** 0-10%.

3.3.1.6 M[PROD][S][0012]: Rapporto linee di commento su linee di codice

- Range di accettazione: $\geq 25\%$;
- Range ottimale: $\geq 30\%$.

3.3.2 M[PROD][S][0013]: Versioni di Android supportate

- Range di accettazione: 1-5;
- Range ottimale: 3-5;

3.3.3 Gamification e framework Octalysis

Sebbene il progetto P2PCS (Peer-To-Peer Car Sharing) sia incentrato sullo sviluppo di un applicazione *Android_{|g|}* di *car sharing_{|g|}*, come suggerisce il suo nome, in realtà una parte importante del progetto è l'implementazione e integrazione di dinamiche di gamification all'interno di un applicativo di questo tipo.

3.3.3.1 Cos'è la gamification

Gamification è un termine che indica l'implementazione di dinamiche tipiche di giochi e videogiochi in ambiti che per loro natura giochi non sono, con l'intento di incentivare l'utente finale a fare cose che non farebbe altrimenti o che non farebbe per un periodo di tempo particolarmente lungo. Sebbene le dinamiche di gamification siano ancora ad uno stadio piuttosto sperimentale, negli ultimi anni essa è stata utilizzata in diversi ambienti (commerciali, pubblicitari, informatici, sociali).

3.3.3.2 Il framework Octalysis

Octalysis è un *framework_{|g|}* che analizza, identifica e organizza in 8 macro categorie chiamate *Core Drives_{|g|}* tutte le tipiche dinamiche implementabili attraverso la gamification. Ogni macro categoria a sua volta è un insieme di sotto-categorie chiamate Components. Core Drives e Components sono definiti in modo generico tali da essere applicabili e identificabili in qualsiasi ambito si intenda ricorrere alla gamification. Per conoscere in modo dettagliato tutti i possibili Components e il loro significato un buon punto di partenza è il sito yukaichou.com, più precisamente le indicazioni presenti all'indirizzo:

<https://yukaichou.com/gamification-examples/>

Per comodo riferimento invece riportiamo di seguito gli 8 Core Drives di Octalysis seguiti da una breve definizione:

1. **Epic Meaning & Calling:** Questo Core Drive riguarda dinamiche per le quali il giocatore si sente come un eletto, un predestinato a qualche grande impresa. Ad esempio il giocatore viene incaricato di creare nuovo materiale e contenuti, oppure viene elargito di un qualche "dono" iniziale.

2. **Development & Accomplishment:** Solitamente è un Core Drive interno, per tutto ciò che riguarda il fare progressi, sviluppare abilità, eventualmente affrontare delle sfide.
3. **Empowerment of Creativity & Feedback:** consiste nel coinvolgere l'utente in processi creativi, a tentare diverse combinazioni e a farlo pensare alle cose che fa. Vengono inoltre utilizzati attentamente meccanismi di feedback, allo scopo di dare all'utente una chiara visione di quello che sta facendo e delle ricompense che ne derivano.
4. **Ownership & Possession:** Questo Core Drive riguarda tutto ciò che dà all'utente la sensazione di possedere qualcosa. L'idea è che se l'utente percepisce di possedere qualcosa da un lato cercherà di far sì che ciò che ha migliori, dall'altro sarà portato a fare qualcosa per possederne ancora di più.
5. **Social Influence & Relatedness:** In questo Core Drive è raggruppato tutto ciò che riguarda la componente di interazione sociale, sia collaborativa (condizione, missioni di gruppo, comunicazione) sia competitiva (invidia, classifiche, gare di gruppo). Ampiamente utilizzata in tempi moderni da moltissime compagnie.
6. **Scarcity & Impatience:** Octalysis definisce questo Core Drive come "volere qualcosa perché non lo puoi avere". Per fare un esempio, un utente vorrebbe avere subito qualcosa che però sarà disponibile solo più tardi nel tempo. L'attesa della cosa aumenta il desiderio per la cosa stessa. Funzionalità relative a questo Core Drive devono però essere implementate con cautela, poiché una loro presenza troppo massiccia potrebbe frustrare l'utente ed allontanarlo dal servizio.
7. **Unpredictability & Curiosity:** Sostanzialmente è il Core Drive che interviene quando l'utente non sa cosa lo aspetta successivamente. Tale meccanismo è fortemente utilizzato, ad esempio, nel gioco d'azzardo, dove la curiosità di vedere come andrà la giocata successiva spinge il giocatore a continuare. Il concetto è applicabile comunque anche ad ambiti più "innocui", come ad esempio la lettura di un libro o la visione di un film, dove vogliamo andare avanti spinti dalla curiosità di sapere cosa accadrà dopo.
8. **Loss & Avoidance:** L'obiettivo di questo Core Drive è di spingere l'utente a compiere determinate azioni per evitare che accada qualcosa di negativo. Ad esempio, evitare penalizzazioni che lo portino a perdere risultati perseguiti fino a questo momento, oppure spingerlo a credere che se non agirà immediatamente perderà l'occasione di farlo in seguito.

3.3.3.3 Gamification applicata al progetto P2PCS Nel nostro caso specifico il progetto prevede di fornire una piattaforma di Car Sharing peer to peer di base (la gestione di particolari complessità che andrebbero previste su una applicazione completa di questo tipo verranno infatti ignorate) sulla quale implementare una serie di idee studiate per i due tipi di utenti cui si rivolgerà l'applicazione: l'utente che affitta un veicolo e l'utente che lo prende in prestito. Per entrambe le categorie

sono sorte diverse domande a cui abbiamo cercato di dare risposta implementando funzionalità relative a specifici Core Drives della gamifications:

- **L'utente che presta l'auto:** Come incentivare l'utente che presta l'auto a farlo spesso, per fasce orarie abbastanza ampie, anche per lunghe tratte? Come far sì che l'utente sia spinto ad affittare il proprio veicolo a più persone diverse possibile, sia nuovi utenti che veterani, a prezzi bassi o possibilmente addirittura gratuitamente?
- **L'utente che chiede un auto:** Come incentivare un utente a sfruttare il servizio di Car Sharing il più possibile, sia per brevi che per lunghe tratte? Come far sì che oltre ad usufruirne egli stesso convinca anche nuovi utenti ad utilizzarlo?

Dopo aver raccolto varie idee e in seguito a numerosi confronti, siamo giunti alle seguenti conclusioni:

- P2PCS farà largo uso dei Core Drives 2 e 5: il primo ponendo costantemente l'utente di fronte ad obiettivi da raggiungere, missioni da concludere e riconoscimenti continui per le varie azioni compiute tramite l'applicazione; il secondo implementando varie componenti social, sia interne all'applicazione (come il continuo confronto e l'interazione con gli altri utenti) sia esterne (come la condivisione delle cose fatte sui social network come Facebook).
- Anche per il Core Drive 4 non è stato eccessivamente approfondito, ma alcuni dei suoi elementi di interazione sono stati reputati interessanti per il progetto.
- Le due idee che sono state maggiormente perseguite nell'implementazione della gamification sono state:
 1. Incentivare l'utente a "diventare qualcuno di importante all'interno della community" guadagnando riconoscimenti per il buon comportamento e l'andare incontro alle necessità degli altri utenti il più possibile.
 2. Spingere l'utente ad usare l'applicazione attraverso incentivi (sia virtuali che economici), convenzioni esterne e riconoscimenti ad uso personale di vario genere.
- I Core Drives 6 e 8 sono i meno presi in considerazione, in quanto ritenuti poco utili o addirittura lesivi per i nostri obiettivi: non è nostra intenzione penalizzare l'utente che non compie azioni, ma trovare un modo positivo e premiante perché lui sia spinto a farle. Inoltre, trattandosi di un'applicazione che si basa su un bisogno dell'utente, ovvero il bisogno di un mezzo di trasporto, si è ritenuto che spingere l'utente ad utilizzarla maggiormente fosse difficilmente "forzabile" con espedienti come quelli del Core Drive 7.

In definitiva l'approccio adottato ai principi della gamification è stato più indirizzato nel dare all'utente una percezione di continuità e miglioramento, come conseguenza dell'utilizzo prolungato dell'applicazione, piuttosto che ad obbligare l'utente a particolari attese o impedire l'accesso ad alcune funzionalità, previo il completamento di specifici obiettivi. Questa scelta è stata motivata dalla natura stessa dell'appli-

cazione, che vede l'utente mettere in gioco il proprio veicolo, ovvero un oggetto di un certo valore economico: la presenza di funzionalità che vengano percepite come limitanti, piuttosto che migliorative, per l'esperienza è stata ritenuta non necessaria.

3.3.3.4 M[PROD][S][0014]: Copertura del framework Octalysis

- **Range di accettazione:** Ciascun Core Drive del framework ha un proprio range di accettazione:
 - **Epic Meaning & Calling:** 1-5;
 - **Development & Accomplishment:** 5-10;
 - **Empowerment of Creativity & Feedback:** 1-5;
 - **Ownership & Possession:** 1-5;
 - **Social Influence & Relatedness:** 3-10;
 - **Scarcity & Impatience:** 0-2;
 - **Unpredictability & Curiosity:** 1-3;
 - **Loss & Avoidance:** 0-2;
- **Range ottimale:** Ciascun Core Drive del framework ha un proprio range ottimale:
 - **Epic Meaning & Calling:** 3-5;
 - **Development & Accomplishment:** 8-10;
 - **Empowerment of Creativity & Feedback:** 3-5;
 - **Ownership & Possession:** 3-5;
 - **Social Influence & Relatedness:** 5-10;
 - **Scarcity & Impatience:** 0-1;
 - **Unpredictability & Curiosity:** 1-2;
 - **Loss & Avoidance:** 0-1;

A Resoconto attività di verifica

In questa sezione verranno riportati in maniera schematica i valori effettivi riscontrati dalla misurazione delle metriche di processo e di prodotto in seguito alle operazioni di verifica della documentazione e del codice precedenti ad ogni revisione del progetto.

A.1 Revisione dei Requisiti

A.1.1 Metriche processi

ID	Valore	Esito	Commento
M[PROC][0001]	0	Ottimo	Tutte le attività definite nel <i>Piano di Progetto v1.0.0</i> sono state completate entro i tempi previsti.
M[PROC][0002]	185€	Ottimo	Come riportato nel consuntivo presente nel <i>Piano di Progetto v1.0.0</i> è stato risparmiato un totale di 185€ rispetto a quanto preventivato.
M[PROC][0003]	0	Ottimo	Nel periodo precedente alla Revisione dei Requisiti non si sono presentati fattori di rischio non previsti dal <i>Piano di Progetto v1.0.0</i> .

Tabella 2: Tabella resoconto dei valori misurati con le metriche di processo

A.1.2 Metriche prodotto

In questo periodo le metriche riguardanti il prodotto che sono state misurate riguardano l'indice di *Gulpease*_[g] per i documenti e la copertura del *framework*_[g] *Octalysis*_[g] basandoci sui requisiti individuati nel documento *Analisi Dei Requisiti v1.0.0*.

A.1.3 M[PROD][D][0001]: Indice di Gulpease

Il seguente grafico riporta, per tutti i documenti approvati ad eccezione del *Glossario v1.0.0*, i risultati delle misurazioni della metrica effettuate tramite lo strumento *Farfalla-Project*_[g], come specificato nelle *Norme di Progetto v1.0.0*.

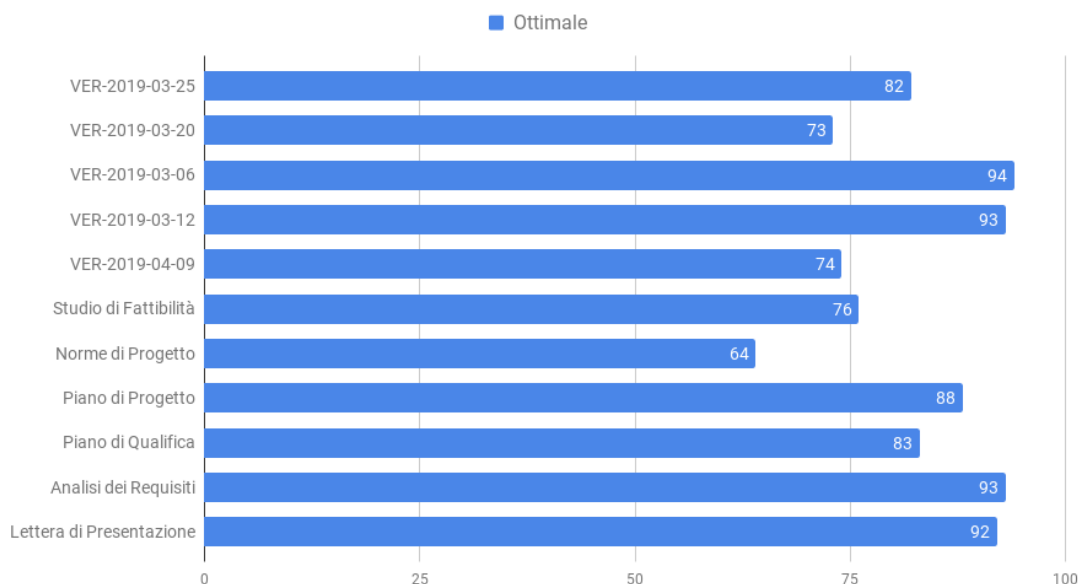


Figura 1: Grafico valori indice Gulpease

A.1.4 M[PROD][S][0014]: Copertura del framework Octalysis

Core Drive abbreviato	Quantità requisiti	Valore	Esito
Meaning	1	1	Accettabile
Accomplishment	15	10	Ottimo
Empowerment	1	1	Accettabile
Ownership	6	4	Ottimo
Social Influence	5	3	Accettabile
Scarcity	0	0	Ottimo
Unpredictability	2	1	Ottimo
Avoidance	0	0	Ottimo

Tabella 3: Tabella resoconto copertura framework Octalysis

Il seguente grafico, ricavato tramite lo strumento online Octalysis-tool, riporta il codice di ogni requisito associato al relativo *Core Drive*_[9]. La dimensione di ogni area, identificata dal nome del Core Drive corrispondente, è proporzionale al valore raggiunto nella metrica sopra riportata.

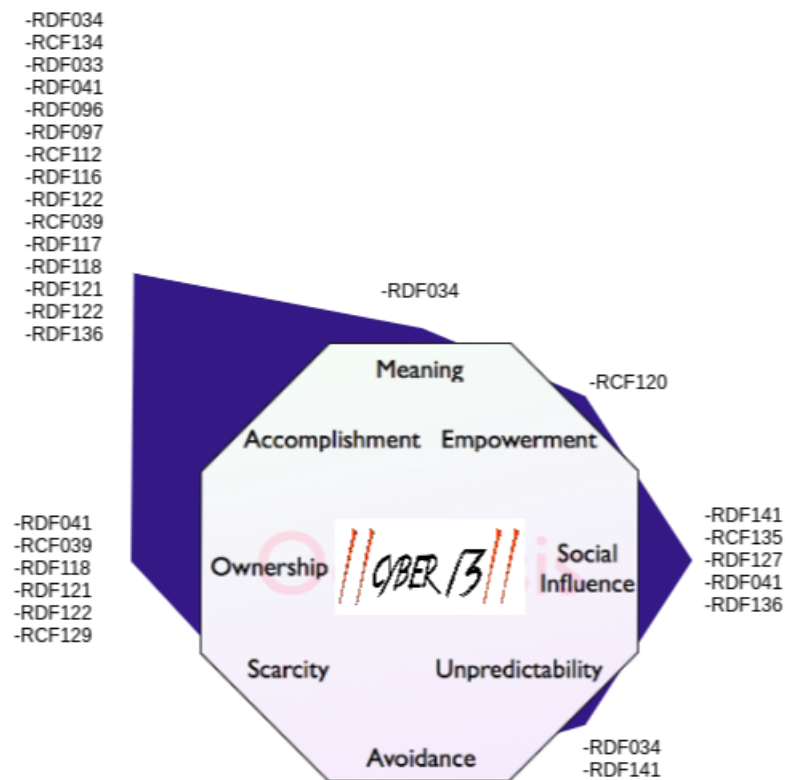


Figura 2: Grafico copertura Core Drives Octalysis