

IronWorks

Utility per la Costruzione di Software Robusto



Swear on Code

Piano di Qualifica

Versione	1.0.0
Redattori	Anna Poletti Sharon Della Libera
Verificatori	Antonio Moz
Responsabili	Francesco Sacchetto
Uso	Esterno
Distribuzione	Gruppo Swear on Code Prof. Tullio Vardanega Prof. Riccardo Cardin Gregorio Piccoli, Zucchetti S.p.A.

Descrizione

Questo documento si occupa di definire le strategie per gestire, controllare e verificare la qualità.

Registro delle modifiche

Descrizione	Autori	Ruolo	Data	Versione
Approvazione	Francesco Sacchetto	Responsabile	2018/04/09	1.0.0
Inserimento Risultati	Sharon Della Libera	Verificatore	2018/04/09	0.1.1
Verifica	Antonio Moz	Verificatore	2018/03/29	0.1.0
Stesura Schema Risultati	Sharon Della Libera	Verificatore	2018/03/28	0.0.7
Modifica Metriche	Anna Poletti	Verificatore	2018/03/28	0.0.6
Stesura Metriche	Sharon Della Libera	Verificatore	2018/03/27	0.0.5
Stesura Obiettivi	Sharon Della Libera	Verificatore	2018/03/26	0.0.4
Stesura Strategie	Anna Poletti	Verificatore	2018/03/23	0.0.3
Stesura Introduzione	Anna Poletti	Verificatore	2018/03/22	0.0.2
Creazione del documento	Sharon Della Libera	Amministratore	2018/03/21	0.0.1

Indice

1	Introduzione	1
1.1	Scopo del documento	1
1.2	Ambiguità	1
1.3	Riferimenti	1
1.3.1	Riferimenti Normativi	1
1.3.2	Riferimenti Informativi	1
2	Strategie	3
2.1	Qualità di processo	3
2.1.1	<i>ISO_G/IEC_G 15504</i>	3
2.1.2	<i>PDCA_G</i>	3
2.2	Qualità di prodotto	3
2.2.1	Modello <i>ISO_G/IEC_G 9126</i>	3
2.3	Oggetto della qualifica	5
2.3.1	Analisi dei Requisiti	5
2.3.2	Processo di Verifica	6
2.3.3	Processo di Documentazione	6
2.3.4	Altre valutazioni	6
3	Obiettivi	7
3.1	Qualità dei processi	7
3.2	Qualità del prodotto	7
3.3	Analisi dei Requisiti	7
3.4	Qualità della documentazione	7
3.5	Qualità di progetto	7
4	Metriche e misure	8
4.1	Qualità processo	8
4.1.1	<i>CMMI_G</i>	8
4.2	Qualità prodotto	9
4.2.1	Software	9
4.2.1.1	Attributi software	9
4.2.2	Qualità dei documenti	9
4.3	Qualità del progetto	9
5	Risultati	11
5.1	Qualità processi	11
5.2	Qualità prodotto	12
5.2.1	Software	12
5.2.2	Documenti	13
5.3	Analisi dei Requisiti	13
5.4	Qualità del progetto	13

Elenco delle tabelle

1	Livelli di qualità del CMMI	8
2	Risultati valutazione interna della maturità	11
3	Valutazione del software secondo gli attributi <i>ISO_G/IEC_G 9126</i>	12
4	Indice di <i>Gulpease_G</i> (ottimo se ≥ 50)	13
5	Requisiti modificati	13
6	Valutazione tempi e costi del progetto	13

1 Introduzione

1.1 Scopo del documento

Questo documento ha lo scopo di descrivere le strategie di verifica, controllo e validazione che il gruppo intende adottare durante lo sviluppo del software per assicurarne la qualità.

1.2 Ambiguità

Al fine di dipanare qualsiasi dubbio o ambiguità relativa al linguaggio impiegato nel documento viene fornito il *Glossario v1.0.0*, documento contenente la definizione di tutti i termini scritti in corsivo e marcati con una 'G' pedice. Per la stesura dei documenti, viene adottato il modello incrementale: dopo ogni revisione si effettuano correzioni e aggiornamenti. Per questo motivo, alcuni processi, attività e/o compiti non verranno trattati nella versione attuale del documento, in quanto inerenti a periodi di sviluppo successivi.

1.3 Riferimenti

1.3.1 Riferimenti Normativi

- *Norme di Progetto v1.0.0*;
- *Capitolato C5 - IronWorks: utilità per la costruzione di software robusto*
<http://www.math.unipd.it/~tullio/IS-1/2017/Progetto/C5.pdf>;
- *ISO/IEC 9001*:
http://www.colonese.it/00-Manuali_Pubblicatii/06-QualitàSoftware_v2.pdf;
- *ISO/IEC 9126*:
http://www.colonese.it/00-Manuali_Pubblicatii/07-ISO-IEC9126_v2.pdf;
- *CMM*:
http://www.colonese.it/00-Manuali_Pubblicatii/12-IntroduzioneCMMI_v1.0.pdf.

1.3.2 Riferimenti Informativi

- Slides del corso di Ingegneria del Software:
 - Qualità di prodotto:
<http://www.math.unipd.it/~tullio/IS-1/2017/Dispense/L13.pdf>;
 - Qualità di processo:
<http://www.math.unipd.it/~tullio/IS-1/2017/Dispense/L15.pdf>.
- *Analisi dei Requisiti v1.0.0*;
- *Piano di Progetto v1.0.0*;

- *Glossario v1.0.0.*

2 Strategie

2.1 Qualità di processo

Con lo scopo di ottenere un prodotto conforme alle attese e di perseguire la qualità di processo, il gruppo ha deciso di adottare lo *standard_G ISO_G/IEC_G 15504*, comunemente denominato *SPICE_G*.

Per un'efficace applicazione dello *standard_G SPICE_G* si è deciso di utilizzare il metodo iterativo *PDCA_G*.

2.1.1 ISO_G/IEC_G 15504

Il modello *ISO_G/IEC_G 15504*, comunemente denominato *SPICE_G*, è un insieme di documenti di *standard_G* tecnici relativi ai processi di sviluppo del software.

ISO_G/IEC_G 15504 contiene un modello di riferimento che stabilisce una dimensione del processo ed una dimensione della capacità.

2.1.2 PDCA_G

Il *PDCA_G*, detto anche "*Ciclo di Deming_G*", è uno schema di gestione iterativo che mira al miglioramento continuo dei processi e dei prodotti in un'ottica a lungo raggio.

Il *PDCA_G* consiste in quattro specifiche fasi:

- **Plan:** stabilire gli obiettivi e i processi adatti al raggiungimento dei risultati attesi;
- **Do:** eseguire ciò che si è pianificato;
- **Check:** studiare i risultati raccolti nella fase precedente e confrontarli con quelli attesi, cioè quelli stabiliti nella fase di pianificazione;
- **Act:** agire in ottica correttiva implementando le soluzioni che hanno superato la fase di controllo.

2.2 Qualità di prodotto

Con lo scopo di garantire la qualità del prodotto il gruppo ha deciso di adottare lo *standard_G ISO_G/IEC_G 9126*.

Il seguente documento fornisce un modello per valutare le qualità esterne, interne e d'uso di un software.

2.2.1 Modello ISO_G/IEC_G 9126

Il modello di qualità del prodotto software descritto dallo *standard_G ISO_G/IEC_G 9126* definisce le caratteristiche e i relativi attributi del software, ciascuna di esse misurabile da metriche interne od esterne.

Il modello consiste in sei caratteristiche:

- **Funzionalità:** rappresenta la capacità del software di fornire tutte le funzioni necessarie per operare in un determinato contesto.
Le sotto-caratteristiche richieste dalla funzionalità sono:

- **Adeguatezza:** capacità di fornire un appropriato insieme di funzioni che permettano agli *utenti_G* di raggiungere gli obiettivi prefissati;
 - **Accuratezza:** capacità di fornire i risultati attesi attenendosi al livello di precisione prefissato;
 - **Interoperabilità:** capacità di interagire con più sistemi;
 - **Sicurezza:** capacità di mantenere protette le informazioni in modo che solamente le persone e i sistemi autorizzati possano accedervi e apportare eventuali modifiche;
 - **Aderenza:** capacità di seguire *standard_G* e convenzioni che siano attinenti con la funzionalità.
- **Affidabilità:** rappresenta la capacità del software di mantenere il livello di prestazione in determinate condizioni indicate.
Le sotto-caratteristiche richieste dall'affidabilità sono:
 - **Maturità:** capacità di evitare che si verifichino errori o risultati non attesi in fase di esecuzione;
 - **Tolleranza ai guasti:** capacità di mantenere il livello di prestazioni in caso di errori;
 - **Recuperabilità:** capacità di ripristinare il livello di prestazioni e di recuperare informazioni in caso di malfunzionamenti;
 - **Aderenza:** capacità di seguire *standard_G* e convenzioni che siano attinenti con l'affidabilità.
 - **Usabilità:** rappresenta la capacità del software di risultare di facile comprensione e studio da parte di un *utente_G* in determinate condizioni.
Le sotto-caratteristiche richieste dall'usabilità sono:
 - **Comprensibilità:** capacità di permettere all'*utente_G* di apprendere la sua funzionalità e di capirne l'utilizzo in determinate condizioni;
 - **Apprendibilità:** capacità di essere facilmente appreso dall'*utente_G*;
 - **Operabilità:** capacità di essere facilmente utilizzabile e controllabile dall'*utente_G*;
 - **Attrattività:** capacità di risultare accattivante all'*utente_G*;
 - **Aderenza:** capacità di aderire a *standard_G* e convenzioni che siano attinenti con l'usabilità.
 - **Efficienza:** rappresenta la capacità del software di realizzare le funzioni richieste nel minor tempo possibile, sfruttando nel miglior modo possibile le risorse disponibili in determinate condizioni.
Le sotto-caratteristiche richieste dall'efficienza sono:
 - **Comportamento rispetto al tempo:** capacità di fornire tempi di risposta ed elaborazione adeguati in determinate condizioni;
 - **Utilizzo delle risorse:** capacità di fare uso di un adeguato numero e tipo di risorse sotto determinate condizioni di utilizzo;

- **Aderenza:** capacità di aderire a *standard_G* e convenzioni che siano attinenti con l'efficienza.
- **Manutenibilità:** rappresenta la capacità del software di essere modificato e corretto. Le sotto-caratteristiche richieste dalla manutenibilità sono:
 - **Analizzabilità:** capacità di poter essere analizzato con lo scopo di individuare errori;
 - **Modificabilità:** capacità di consentire l'implementazione di modifiche al software originale;
 - **Stabilità:** capacità di evitare effetti indesiderati a seguito di modifiche;
 - **Provabilità:** capacità di eseguire i test al prodotto modificato;
 - **Aderenza:** capacità di aderire a *standard_G* e convenzioni che siano attinenti con la manutenibilità.
- **Portabilità:** rappresenta la capacità del software di poter essere trasportato da un ambiente ad un altro. Le sotto-caratteristiche richieste dalla portabilità sono:
 - **Adattabilità:** capacità di poter essere adattato a differenti ambienti senza la necessità di azioni non previste dal prodotto;
 - **Installabilità:** capacità di essere installato in un determinato ambiente;
 - **Coesistenza:** capacità di condividere risorse con altre applicazioni;
 - **Sostituibilità:** capacità di prendere il posto di un altro software per lo stesso scopo e nello stesso ambiente;
 - **Aderenza:** capacità di aderire a *standard_G* e convenzioni che siano attinenti con la portabilità.

2.3 Oggetto della qualifica

Aderendo ai sopracitati *standard_G*, il gruppo mira a pianificare, controllare e verificare i vari processi, attività e compiti presenti nei periodi di sviluppo in esame.

2.3.1 Analisi dei Requisiti

Durante l'attività *Analisi dei Requisiti* del *Processo di Sviluppo* è necessario perseguire la completezza e correttezza dell'interpretazione dei requisiti.

Nelle *Norme di Progetto v1.0.0* sono definite le caratteristiche principali che un requisito deve avere e i formalismi che deve adottare. Viene spiegato inoltre come strutturare la tabella del tracciamento dei requisiti stessi, fondamentale per avere sotto controllo questo processo. Il gruppo si impegna a rispettare tali regole.

Per misurare la qualità del processo è necessario tenere traccia del numero di errori rilevati durante l'intero ciclo di sviluppo, imputabili ad una incompleta o non corretta interpretazione dei requisiti.

Quindi meno errori e modifiche si introdurranno nelle fasi successive, più questo processo può definirsi di qualità.

2.3.2 Processo di Verifica

In questo processo si garantisce l'efficacia dei test.

Per l'attuale fase di sviluppo del progetto si può testare l'efficacia del livello di copertura dei test rispetto ai requisiti, analizzando la tabella della tracciabilità nell'*Analisi dei Requisiti v1.0.0*.

Facciamo riferimento a *PragmaDB_G*, strumento descritto nelle *Norme di Progetto v1.0.0*, che ci aiuta a gestire i requisiti e ad evitare di tralasciarne qualcuno.

2.3.3 Processo di Documentazione

Garantire la qualità della documentazione prodotta è fondamentale, in quanto essa è il mezzo formale con cui *fornitori_G* e *committenti_G* si rapportano per quanto riguarda il prodotto software.

Si misura:

- Verificando la completezza della documentazione, ovvero se vengono consegnati tutti i documenti previsti dal *committente_G*;
- Verificando se tali documenti rispettano le norme definite nelle *Norme di Progetto v1.0.0* che il gruppo si impegna a rispettare;
- Utilizzando le misure e le metriche definite alla fine di questo documento.

2.3.4 Altre valutazioni

Vengono inoltre definite le metriche per valutare la qualità del progetto, misurando i risultati ottenuti con quelli pianificati nel *Piano di Progetto v1.0.0*.

Lo scopo è quello di definire quantitativamente la capacità del gruppo di rispettare tempi e costi.

3 Obiettivi

Il gruppo, in riferimento alle metriche indicate nell'ultima sezione di questo documento, si fissa alcuni obiettivi "quantitativi" per perseguire la qualità.

3.1 Qualità dei processi

Utilizzando lo schema nella tabella 1, il gruppo si pone l'obiettivo ottimale di raggiungere il livello "Ottimizzato" di maturità della scala del $CMMI_G$.

Tuttavia si ritiene accettabile il livello "Gestito quantitativamente".

3.2 Qualità del prodotto

Seguendo la tabella 3 a fine documento, si riportano le valutazioni riguardanti le caratteristiche del software.

L'obiettivo ottimale è avere l'80% delle valutazioni "Alta".

Si ritiene comunque accettabile ottenere l'80% delle valutazioni "Media".

3.3 Analisi dei Requisiti

Durante tutto lo sviluppo del progetto l'aggiunta, la modifica o l'eliminazione di alcuni requisiti non obbligatori è un'attività lecita, in quanto possono sorgere nuove esigenze in corso d'opera.

Il $rischio_G$ è quello di definire male all'inizio i requisiti e quindi doverli modificare pesantemente, avendo ripercussioni su tutte le attività che vengono sviluppate parallelamente.

Pertanto è d'obbligo una buona analisi e la modifica del 30% dei requisiti nel complesso determina un livello di qualità "Alto" per questo processo, obiettivo del gruppo.

Si considera livello di qualità "Medio" fino al 50% dei requisiti modificati e "Basso" per percentuali superiori.

3.4 Qualità della documentazione

Con lo scopo di dotare i documenti di una buona leggibilità, il gruppo ha deciso di ritenere accettabile il seguente range di qualità:

- **Indice $Gulpease_G$:** [50 | 100].

3.5 Qualità di progetto

Con lo scopo di controllare qualitativamente il progetto, il gruppo ha deciso di ritenere accettabili i seguenti valori:

- **$Schedule\ Variance_G$:** valore ≥ 0 ;
- **$Budget\ Variance_G$:** valore ≥ 0 .

4 Metriche e misure

4.1 Qualità processo

4.1.1 $CMMI_G$

Il gruppo decide di misurare la qualità dei processi prendendo a riferimento le *best practices_G* definite nel $CMMI_G$ (Capability Maturity Model Integration), approccio adottato in contesti aziendali.

Il gruppo decide di acquisire le *best practices_G* e adattare nell'ambito di questo progetto. Pertanto i livelli di maturità da noi identificati non hanno un effettivo riscontro con quelli ufficiali, ma sono costruiti appositamente per creare una scala di maturità conforme alla dimensione e alle capacità del gruppo.

La tabella 1 fornisce una guida per la qualità dei processi e un punto di riferimento per la loro valutazione.

Livello	Processi da ottimizzare
Iniziale	Nessun processo
Gestito	Gestione dei requisiti
	Pianificazione del progetto software
	Controllo del progetto software
	Controllo della qualità del software
	Gestione della configurazione del software
Definito	Attenzione al processo organizzativo
	Definizione del processo organizzativo
	Gestione dell'integrazione del software
	Ingegnerizzazione del software
	Coordinamento all'interno del gruppo
Gestito quantitativamente	Gestione degli aspetti quantitativi dei processi
	Gestione della qualità dei software
Ottimizzato	Prevenzione dei difetti
	Gestione delle modifiche tecnologiche
	Gestione delle modifiche ai processi

Tabella 1: Livelli di qualità del CMMI

4.2 Qualità prodotto

4.2.1 Software

4.2.1.1 Attributi software

Si definiscono le metriche per valutare il livello di qualità del software prodotto in riferimento agli attributi definiti dal $CMMI_G$.

- Valutazione "Alta";
- Valutazione "Media";
- Valutazione "Bassa";
- Valutazione "Non valutabile" (NV);
- Valutazione "Da valutare" (DV).

4.2.2 Qualità dei documenti

Al fine di raggiungere una buona leggibilità dei documenti il gruppo ha deciso di applicare la seguente metrica:

- **Indice $Gulpease_G$** : misura la leggibilità di un testo, individuandone la complessità dello stile. L'indice è tarato sulla lingua italiana ed ha il vantaggio di utilizzare la lunghezza delle parole in lettere invece che in sillabe, semplificando il procedimento di calcolo automatico.

L'indice di $Gulpease_G$ considera due variabili: la lunghezza della parola e la lunghezza della frase rispetto al numero delle lettere.

La formula di calcolo è la seguente:

$$89 + \frac{300 \times (\text{numero-delle-frasi}) - 10 \times (\text{numero-delle-lettere})}{(\text{numero-delle-parole})}$$

I risultati sono compresi tra 0 (leggibilità più bassa) e 100 (leggibilità più alta);

4.3 Qualità del progetto

Al fine di controllare qualitativamente il progetto, il gruppo ha deciso di applicare le seguenti metriche:

- **$Schedule\ Variance_G$** : è un indicatore di efficacia, il quale mostra se si è o meno coerenti con la pianificazione temporale rispetto alle attività nella $baseline_G$.

Una $schedule\ variance_{GG}$ positiva indica che il gruppo è in anticipo rispetto a quanto stabilito nel Piano di Progetto.

Una $schedule\ variance_G$ negativa indica che il gruppo è in ritardo rispetto a quanto stabilito nel Piano di Progetto.

La formula è la seguente:

$$SV = BCWP - BCWS$$

Dove:

- **BCWP (Budgeted Cost of Work Perormed)**: valore delle attività realizzate alla data corrente in termini di giorni;
 - **BCWS (Budgeted Cost of Work Scheduled)**: valore delle attività pianificate alla data corrente in termini di giorni.
- **Budget Variance_G**: indica se la spesa sostenuta in una determinata data supera o no quella preventivata.
Una *budget variance_G* positiva indica che si è speso meno di quanto inizialmente previsto nel Piano di Progetto.
Una *budget variance_G* negativa indica che si è speso più di quanto inizialmente previsto nel Piano di Progetto.
La formula è la seguente:

$$BV = BCWS - ACWP$$

Dove:

- **BCWS (Budgeted Cost of Work Scheduled)**: costo pianificato per la data corrente valutato in Euro;
- **ACWP (Actual Cost of Work Perormed)**: costo effettivamente sostenuto alla data corrente valutato in Euro.

5 Risultati

Di seguito si riportano i risultati delle valutazioni effettuate.
La valutazione si effettua alla fine di ogni revisione.

5.1 Qualità processi

Revisione	Livello	Superato
RR	Livello Gestito	Non ancora superato
RP		
RQ		
RA		

Tabella 2: Risultati valutazione interna della maturità

5.2 Qualità prodotto

5.2.1 Software

Caratteristica	Attributi	RR	RP	RQ	RA
Funzionalità	Adeguatezza	DV			
	Accuratezza	DV			
	Interoperabilità	DV			
	Sicurezza	DV			
	Aderenza	DV			
Affidabilità	Maturità	DV			
	Tolleranza ai guasti	DV			
	Recuperabilità	DV			
	Aderenza	DV			
Usabilità	Comprensibilità	DV			
	Apprendibilità	DV			
	Operabilità	DV			
	Attrattività	DV			
	Aderenza	DV			
Efficienza	Comportamento rispetto al tempo	DV			
	Utilizzo risorse	DV			
	Aderenza	DV			
Manutenibilità	Analizzabilità	DV			
	Modificabilità	DV			
	Stabilità	DV			
	Provabilità	DV			
	Aderenza	DV			
Portabilità	Adattabilità	DV			
	Installabilità	DV			
	Coesistenza	DV			
	Sostituibilità	DV			
	Aderenza	DV			

Tabella 3: Valutazione del software secondo gli attributi *ISO_G/IEC_G 9126*

5.2.2 Documenti

Documento	RR	RP	RQ	RA
Analisi dei Requisiti	87			
Glossario	56			
Norme di Progetto	79			
Piano di Qualifica	65			
Piano di Progetto	67			
Studio di Fattibilità	67			

Tabella 4: Indice di $Gulpease_G$ (ottimo se ≥ 50)

5.3 Analisi dei Requisiti

Revisione	Percentuale requisiti modificati	Superato
RR	Stesura di 83 requisiti	Non ancora modificati
RP		
RQ		
RA		

Tabella 5: Requisiti modificati

5.4 Qualità del progetto

Revisione	BCWP (Giorni)	BCWS (Giorni)	BCWS (Euro)	ACWP (Euro)	SV	BV	Superato
RR	35	35	4525,00	4440,00	0	85,00	Superato
RP							
RQ							
RA							

Tabella 6: Valutazione tempi e costi del progetto