



Glossario

7DOS - 5 Aprile 2019

Informazioni sul documento

Versione	3.0.0
Responsabile	Giovanni Sorice
Verifica	Michele Roverato Marco Costantino
Redazione	Giacomo Barzon Lorenzo Busin
Stato	Approvato
Uso	Esterno
Destinato a	Prof. Tullio Vardanega Prof. Riccardo Cardin Zucchetti 7DOS
Email	7dos.swe@gmail.com

Descrizione

Questo documento rappresenta il glossario dei termini utilizzati nella documentazione di progetto *G&B*.

Diario delle modifiche

Versione	Data	Descrizione	Autore	Ruolo
3.0.0	2019-04-05	<i>Approvazione del documento per rilascio RQ</i>	Giovanni Sorice	Responsabile
2.3.0	2019-04-04	<i>Verifica del documento</i>	Michele Roverato	Verificatore
2.2.0	2019-04-03	<i>Verifica del documento</i>	Marco Costantino	Verificatore
2.1.1	2019-03-25	<i>Stesura termini</i>	Giacomo Barzon	Amministratore
2.1.0	2019-03-19	<i>Stesura termini</i>	Lorenzo Busin	Amministratore
2.0.0	2019-03-05	<i>Approvazione del documento per rilascio RP</i>	Marco Costantino	Responsabile
1.4.0	2019-03-04	<i>Verifica del documento</i>	Lorenzo Busin	Verificatore
1.3.0	2019-03-02	<i>Verifica del documento</i>	Giovanni Sorice	Verificatore
1.2.3	2019-02-25	<i>Stesura termini</i>	Nicolò Tartaggia	Amministratore
1.2.2	2019-02-12	<i>Stesura termini</i>	Michele Roverato	Analista
1.2.1	2019-02-10	<i>Stesura termini</i>	Michele Roverato	Analista
1.2.0	2019-01-29	<i>Verifica del documento</i>	Giacomo Barzon	Verificatore
1.1.2	2019-01-26	<i>Stesura termini</i>	Andrea Trevisin	Analista
1.1.1	2019-01-21	<i>Stesura termini</i>	Andrea Trevisin	Analista
1.0.0	2019-01-10	<i>Approvazione del documento per rilascio RR</i>	Lorenzo Busin	Responsabile
0.4.0	2019-01-09	<i>Verifica del documento</i>	Michele Roverato	Verificatore
0.3.0	2019-01-07	<i>Verifica del documento</i>	Giovanni Sorice	Verificatore
0.2.0	2019-01-05	<i>Verifica del documento</i>	Giacomo Barzon	Verificatore
0.1.6	2019-01-04	<i>Stesura termini</i>	Marco Costantino	Analista

Versione	Data	Descrizione	Autore	Ruolo
0.1.5	2018-12-29	<i>Stesura termini</i>	Marco Costantino	Analista
0.1.4	2018-12-21	<i>Stesura termini</i>	Nicolò Tartaggia	Amministratore
0.1.3	2018-12-18	<i>Stesura termini</i>	Andrea Trevisin	Analista
0.1.2	2018-12-13	<i>Stesura termini</i>	Andrea Trevisin	Analista
0.1.1	2018-12-10	<i>Stesura termini</i>	Marco Costantino	Analista
0.1.0	2018-12-05	<i>Stesura termini</i>	Nicolò Tartaggia	Amministratore
0.0.1	2018-12-05	<i>Stesura dello scheletro del documento</i>	Andrea Trevisin	Analista

Indice

A	7
Alert	7
Analisi dei requisiti	7
Airbnb CSS	7
Analisi dinamica	7
Analisi statica	7
Angular TypeScript	7
Astah UML	8
Attività	8
B	9
Baseline	9
Best Practice	9
Bottom-Up	9
Brainstorming	9
Budget Variance	9
Build	9
C	10
Capitolato d'appalto	10
Camel case	10
Ciclo di vita del software	10
Cloud	10
CoCoMo	10
Code-'n-Fix	11
Commit	11
Committente	11
Compito	11
Conditional Probability Table(CPT)	11
Configurazione	11
Controllo di configurazione	11
Coveralls	12
CSS	12
D	13
Dashboard	13
Design Pattern	13
DevOps	13
Diff	13
Disciplinato	13
E	14
Economicità	14
Efficacia	14
Efficienza	14

F	15
Fault	15
G	16
Grafana	16
Grafana Plug-in Code Style Guide	16
H	17
HTML	17
HTML5 Style Guide and Coding Conventions	17
I	18
IEE 830-1998	18
Incremento	18
InfluxDB	18
Reverse Engineering	18
Intelligenza artificiale	18
In-Schedule	18
ISO/IEC 12207	18
ISO/IEC 15504	19
ISO/IEC 25010	19
ISO/IEC/IEEE 29119	19
ISO/IEC/IEEE 42010:2011	19
ISO 90003:2004	20
Issue Tracking System	20
Iterazione	20
J	21
JavaScript	21
JavaScript Style Guide and Coding Conventions	21
JPG	21
JSON	21
L	22
LaTeX	22
Likelihood	22
Lint	22
Liveliness	22
M	23
Management	23
Manuale della qualità	23
Manutenzione	23
Miglioramento continuo	23
Milestone	23
Minimizzazione	23
Modello di ciclo di vita	23

Sequenziale (a cascata)	24
Incrementale	24
Evolutivo	24
A componenti	24
A spirale	24
Agili	24
Scrum	24
Modulo	25
Monitoraggio dei rischi	25
O	26
Obiettivo SMART	26
Open source	26
P	27
Panel	27
PDCA	27
Pianificazione delle attività	27
Piano di progetto	27
Plug-in	28
PNG	28
Processo	28
Processo software	28
Processo specializzato per progetto	28
Prodotto software	28
Prodotti documentali	29
Produttività	29
Progettazione software	29
Progetto	29
Project manager	29
Proponente	29
Prototipo	29
Q	30
Qualità	30
Qualità del prodotto	30
Qualità dei processi	30
Quantificabile	30
R	31
Raintank	31
Repository	31
Requisito	31
Rete Bayesiana	31
Rischio	31
Riuso	32
Rule Declaration	32

Ruolo	32
S	34
SASS	34
SASS Style Guide	34
Schedule Variance	34
Script	34
Sistematico	34
Slack (tempo)	34
Software Engineering	34
Sorgente Dati	35
Stima dei costi	35
Studio di fattibilità	35
T	36
Task	36
Ticket	36
Tracciamento dei requisiti	36
TravisCI	36
Trender	36
TSLint	36
TypeScript	36
V	37
Validare	37
Verificare	37
Verificatore di requisiti	37
Versionamento	37
Versione	37
W	38
Way of working	38
Webhook	38
Workflow	38
X	39
XML	39
Z	40
Zero-latency	40
Zero-laxity	40

A

Alert

Notifica.

Analisi dei requisiti

L'obiettivo dell'analisi dei requisiti è individuare e definire i requisiti di progetto. Il risultato di tale processo è un documento contrattuale che andrà fornito all'azienda appaltatrice nell'ambito di una gara d'appalto. Sulla base di questo documento l'azienda deciderà a chi affidare l'appalto.

Airbnb CSS

Insieme di convenzioni da rispettare per la produzione di codice CSS e SASS di qualità, reperibile al seguente link [Airbnb Style guide](#).

Analisi dinamica

Tecnica di analisi di un prodotto software in cui quest'ultimo viene eseguito in un ambiente reale o virtuale; coincide con i test:

- **Di unità:** verificano il corretto funzionamento dell'unità;
- **Di integrazione:** verifica la corretta interazione tra più unità che interagiscono dando luogo ad un *build*;
- **Di sistema:** verificano la corretta interazione tra più *build* che compongono un sistema complesso.

Il collaudo, test finale eseguito con il committente, è solo un'attività formale se i test precedenti sono stati eseguiti rigorosamente. I test devono essere ripetibili e deterministici, perciò vanno documentati: le condizioni iniziali di esecuzione, l'esecuzione, il risultato atteso ed il risultato ottenuto. I test inoltre devono essere automatizzati per ridurre al minimo la possibilità di errore umano.

Analisi statica

Tecnica di analisi di un prodotto software in cui questo non viene eseguito, bensì ne viene esaminato il codice. *Walkthrough* e *Inspection* sono le due tecniche di analisi statica prevalenti: *Walkthrough* prevede un'analisi a tappeto del codice sorgente alla ricerca di possibili criticità, mentre *Inspection* somiglia di più all'analisi a campione e prevede l'analisi specifica di parti considerate di importanza critica.

Angular TypeScript

Framework costruito interamente in TypeScript.

Astah UML

Astah UML è uno strumento di progettazione software gratuito che supporta gli ultimi diagrammi UML. La generazione del codice è disponibile con plug-in gratuiti.

Attività

L'attività è una componente essenziale di un progetto. Un'attività prevede dell'intenzionalità. Le specifiche dell'attività sono determinate da chi la svolge.

B

Baseline

La *baseline* è in generale un punto di partenza, il piano di progetto originale. Essa ha diverse declinazioni che indirizzano un obiettivo strategico e il cui scopo è aiutare a misurare l'avanzamento del processo nella direzione degli obiettivi prefissati. Questi vengono concordati con il committente, in modo di dimostrare l'avanzamento del progetto. La *baseline* è suddivisa in parti, definite nel modo migliore per aiutare a raggiungere gli obiettivi. Ogni parte evolve nel tempo ed ha quindi un numero di versioni, a cura dell'*owner* (responsabile) della parte. Il mantenimento della *baseline* avviene tramite la gestione della configurazione.

Best Practice

Una *best practice* è il modo migliore di approcciare un problema. Le *best practice* prevedono l'applicazione di principi noti ed autorevoli. In ingegneria è richiesto conoscere ed usare le *best practice* esistenti, non crearne di nuove.

Bottom-Up

Tecnica di sviluppo che consiste nel comporre sistemi complessi partendo da moduli di dimensioni e complessità inferiori.

Brainstorming

Discussioni collaborative e creative, in cui viene data voce ad ogni persona a turno. Hanno l'intento di far emergere più spunti e concetti possibili, per poi operare una selezione di quelli più interessanti. Per garantire che idee e considerazioni non vadano perse dev'essere designato un membro per la stesura di un verbale.

Budget Variance

La *Budget Variance* è una metrica che indica se la spesa *complessiva* attuale è maggiore o minore alla spesa pianificata. Viene calcolata: $BV = EV - AC$, dove BV indica la *budget variance*, EV indica l'*earned value*, ovvero il valore delle attività realizzate, AC indica l'*actual cost*, ovvero il costo effettivamente sostenuto.

Build

Processo di trasformazione del codice sorgente in software eseguibile. Tramite alcuni strumenti, come TravisCi, è possibile automatizzarlo.

C

Capitolato d'appalto

Documento, prodotto da un'azienda, che descrive un prodotto o sistema che questa desidera sia realizzato. Il capitolato è una chiamata ai fornitori, una notifica di bisogno che andrà esaminata per verificare se è il caso di partecipare al bando d'appalto per la realizzazione del prodotto richiesto. Costituisce la prima fonte di informazioni da analizzare per l'analisi dei requisiti. Il dominio dell'azienda appaltatrice influenzerà le informazioni contenute nel capitolato che quindi dovrà essere attentamente analizzato tenendo conto del contesto professionale da cui arriva; sarà necessario un dialogo con l'azienda appaltatrice per la corretta comprensione del capitolato.

Camel case

La notazione a cammello o in inglese Camel case è la pratica nata durante gli anni settanta di scrivere parole composte o frasi unendo tutte le parole tra loro, ma lasciando le loro iniziali maiuscole. Il nome (letteralmente "carattere a cammello") deriva dai "salti" all'interno di una parola, che fanno venire in mente le gobbe di un cammello. La prima lettera può essere sia maiuscola (es. CamelCase), come il nome delle classi, che minuscola (es. camelCase), come le proprietà di un oggetto.

Ciclo di vita del software

In ingegneria del software, modo in cui l'attività di realizzazione di un prodotto software viene scomposta in sotto-attività coordinate tra loro, e il cui risultato è il prodotto software e tutta la documentazione necessaria. Il ciclo di vita del software varia a seconda della metodologia di sviluppo adottata.

Cloud

Abbreviativo di "*cloud computing*" ("computazione sulle nuvole"). Indica un paradigma di erogazione di servizi informatici da un fornitore ad un cliente attraverso Internet. In particolare, il fornitore permette al cliente di accedere e configurare delle risorse fisiche in remoto, *on demand*.

CoCoMo

Modello algoritmico per la stima di costi e risorse. Esso stima le risorse necessarie e le esprime in mesi/persona (MP). Ha come input:

- La complessità del progetto;
- Le dimensioni del SW da sviluppare;
- Il peso della complessità sullo sviluppo;
- Un coefficiente moltiplicativo (parte da 1);

- Un fattore di espansione del tempo (parte da 2.5);
- Un coefficiente di complessità.

Code-'n-Fix

Vecchia pratica nell'ambito di produzione del software, autodescrittiva: l'intera attività di produzione consisteva nel codificare e riparare software.

Commit

Insieme di diff che sono state esplicitamente validate. Ogni commit produce una nuova versione del codice sorgente contenuto all'interno della Repository.

Committente

Figura che commissiona un lavoro.

Compito

Un compito è una componente essenziale di un progetto. I compiti vengono assegnati e non lasciano spazio alla decisione di chi li riceve.

Conditional Probability Table(CPT)

Una Conditional Probability Table (CPT) è una tabella in cui:

- Ogni colonna rappresenta l'i-esimo stato in cui il nodo corrente può risiedere. Ogni stato è identificato da un nome ed un intervallo di valori associato. Un nodo non può possedere più stati con intervalli di valori sovrapposti tra loro;
- Ogni riga rappresenta la j-esima combinazione esistente di tutti i possibili stati dei nodi predecessori;
- le celle interne indicano la probabilità, condizionata dai nodi predecessori, che il nodo corrente si trovi in uno specifico stato data una combinazione esistente dei possibili stati dei nodi predecessori.

Configurazione

Insieme di regole che determina come assemblare le sezioni di un software, quali versioni usare per ogni sezione, come le sezioni interagiscono, quali sezioni con quali versioni producono una baseline , etc. etc..

Controllo di configurazione

Gestione e controllo della configurazione che permette di assemblare le varie componenti di un software.

Coveralls

Coveralls è un servizio web che permette di monitorare la copertura dei test effettuati sul codice sorgente depositato all'interno di repository GitHub, Bitbucket o GitLab.

CSS

Il CSS, acronimo di Cascading Style Sheets, è un linguaggio usato per definire la formattazione di documenti HTML.

D

Dashboard

Una dashboard (in italiano "cruscotto") è una schermata che permette di monitorare in tempo reale l'andamento dei report e delle metriche aziendali. Una dashboard può presentare uno o più panel al suo interno.

Design Pattern

Un design pattern è una soluzione generica ad un problema ricorrente. I design pattern non sono specifici ad un linguaggio di programmazione e individuano piuttosto l'approccio ottimale in contesto ben definito per la risoluzione di un problema generico. Quando si intende usare un design pattern è quindi necessario istanziarlo, ovvero è necessario implementarlo in modo appropriato rispetto ai requisiti del sistema da realizzare e al linguaggio di programmazione usato.

DevOps

Termine derivante dalla contrazione delle parole inglesi *development* ("sviluppo") e *operations* ("operazioni" nel senso di "messa in produzione"). Indica un metodo di sviluppo software basato sulla comunicazione e collaborazione tra sviluppatori e addetti alle "*IT operations*", per sviluppare e mantenere velocemente ed efficacemente prodotti e servizi software.

Diff

Insieme di linee di codice modificate su un singolo file.

Disciplinato

Soggetto ad un insieme di regole pensate per garantire la massima efficienza ed efficacia.

E

Economicità

Miglior compromesso tra Efficacia ed efficienza. Garantita dall'uso di standard.

Efficacia

L'abilità di un'entità di portare a termine il compito assegnatole.

Efficienza

Data la misura del consumo di risorse che avviene nel compimento di un obiettivo, minore il consumo di risorse, maggiore è l'efficienza.

F

Fault

Condizione anormale o un difetto di un elemento, dell'attrezzatura oppure del sottosistema che possono condurre ad un guasto o un fallimento del sistema stesso.

G

Grafana

Grafana è un compositore di grafici, che viene eseguito come applicazione web. Supporta InfluxDB.

Grafana Plug-in Code Style Guide

Guida tecnica per la scrittura di codice per plug-in di Grafana.

H

HTML

In informatica l'HyperText Markup Language (HTML) è un linguaggio di markup nato per la formattazione e impaginazione di documenti ipertestuali disponibili nel web. HTML è un linguaggio di pubblico dominio, la cui sintassi è stabilita dal *World Wide Web Consortium* (W3C).

HTML5 Style Guide and Coding Conventions

Insieme di convenzioni da rispettare, definite da W3c per la produzione di codice HTML di qualità, reperibile al seguente link [HTML Style guide](#)

I

IEE 830-1998

Best practice raccomandate per la specifica dei requisiti di prodotti software.

Incremento

Procedura che prevede avanzamento per aggiunta ad un impianto base.

InfluxDB

InfluxDB è un database di serie temporali open source. È ottimizzato per la memorizzazione e il recupero rapido e ad alta disponibilità di dati di serie temporali in campi come il monitoraggio delle operazioni, le metriche dell'applicazione, i dati di sensori e l'analisi in tempo reale.

Reverse Engineering

Il processo di reverse engineering (anche chiamato in italiano ingegneria inversa) consiste nell'analisi dettagliata del funzionamento, progettazione e sviluppo di un oggetto (dispositivo, componente elettrico, meccanismo o software) al fine di produrre un secondo oggetto che abbia un funzionamento analogo, magari migliorando o aumentando l'efficienza dello stesso, senza in realtà copiare niente dall'originale; inoltre, si può tentare di realizzare un secondo oggetto in grado di interfacciarsi con l'originale.

Intelligenza artificiale

Disciplina nel ramo dell'informatica, si occupa dello studio e della progettazione di sistemi informatici in grado di funzionare in maniera simile alla mente umana. Per raggiungere questo obiettivo, si occupa anche di elaborare modelli matematici che approssimino le modalità di pensiero ed apprendimento, caratteristiche dei cervelli biologici.

In-Schedule

Il termine in-schedule indica la conformità alle aspettative temporali o di costi determinate in fase di pianificazione.

ISO/IEC 12207

Standard riferiti ai processi di ciclo di vita, raggruppati in 3 categorie: primari, di supporto, organizzativi.

ISO/IEC 15504

Standard riferiti alla qualità dei processi, collettivamente noti anche come SPICE (*Software Process Improvement and Capability Determination*). Definiscono una scala di maturità di processo a cinque livelli (più il livello base, detto "livello 0").

ISO/IEC 25010

Standard riferito alla qualità di prodotto, noto anche come SQuaRE (*Systems and software Quality Requirements and Evaluation*). Descrive un modello di qualità di prodotto che individua 8 caratteristiche principali da rispettare:

- **Functional Suitability:** idoneità funzionale, ossia quanto il prodotto rispetta i requisiti;
- **Performance Efficiency:** efficienza della performance, ossia quanto il prodotto è in grado di sfruttare efficientemente le risorse disponibili;
- **Compatibility:** compatibilità, ossia quanto il prodotto riesce ad interfacciarsi con altri prodotti o sistemi;
- **Usability:** usabilità, ossia quanto il prodotto risulta di semplice utilizzo;
- **Reliability:** affidabilità, ossia quanto il prodotto risulta stabile nell'esecuzione e nei risultati;
- **Security:** sicurezza, ossia quanto il prodotto protegge i dati trattati;
- **Maintainability:** manutenibilità, ossia quanto il prodotto si presta alla risoluzione di bug e problemi;
- **Portability:** portabilità, ossia quanto il prodotto risulta utilizzabile su ambienti diversi tra loro.

ISO/IEC/IEEE 29119

Serie di cinque standard internazionali che regolamentano le procedure per il software testing.

ISO/IEC/IEEE 42010:2011

Standard di best practice concernenti la progettazione software e la definizione dell'architettura del software. Punti essenziali sono:

- La decomposizione del sistema in componenti (utile ad aumentare il parallelismo);
- Definizione delle interfacce dei componenti;
- Definizione dell'organizzazione delle interfacce che permettono l'interazione dei componenti;
- Paradigmi vari per la composizione dei componenti.

ISO 9003:2004

Standard di best practice per la valutazione della qualità di processi dei fornitori. I principi fondamentali sono:

- L'orientamento al cliente;
- L'obiettivo di leadership sul mercato;
- Il coinvolgimento del personale;
- L'approccio per processi;
- L'obiettivo del miglioramento continuo;
- La presa di decisioni basate su evidenze;
- La gestione delle relazioni.

A garantire l'adesione a questi principi dev'essere la documentazione verticale (specifica di progetto) ed orizzontale (specifica dell'azienda).

Issue Tracking System

Prodotto software che gestisce e mantiene liste di problematiche. Utile in situazioni di sviluppo collaborativo da parte di un team.

Iterazione

Procedura che prevede avanzamento per raffinamento e rivisitazioni.

J

JavaScript

Linguaggio di scripting orientato agli oggetti e agli eventi, comunemente utilizzato nella programmazione web, lato client, per la creazione di effetti dinamici interattivi tramite funzioni di script, invocate da eventi innescati a loro volta in vari modi dall'utente sulla pagina web in uso.

JavaScript Style Guide and Coding Conventions

Insieme di convenzioni da rispettare, definite da W3C, per la produzione di codice JavaScript di qualità, reperibile al seguente link [JavaScript Style guide](#).

JPG

Formato lossy per memorizzare immagini.

JSON

JSON(JavaScript Object Notation) è un formato adatto all'interscambio di dati fra applicazioni client/server basato sul linguaggio JavaScript.

L

LaTeX

Linguaggio di markup per la scrittura di testi, particolarmente usato in ambito accademico.

Likelihood

Parola inglese, significa "verosimiglianza" in italiano. In ambito probabilistico si usa per indicare la probabilità che un evento si verifichi.

Linter

Strumento che analizza il codice sorgente e segnala eventuali errori, bug, errori stilistici e costrutti sospetti.

Liveliness

Parola inglese, significa "vitalità" in italiano. In ambito informatico si riferisce ad un insieme di proprietà dei sistemi concorrenti, per cui un sistema deve funzionare nonostante i suoi componenti concorrenti debbano alternarsi in sezioni critiche.

M

Management

L'insieme delle funzioni amministrative, direttive e gestionali di un'impresa o di un'azienda; il compito del manager.

Manuale della qualità

Documento che specifica le strategie che un organizzazione adotta per operare processi di qualità.

Manutenzione

Processo correttivo e di sviluppo che avviene dopo il rilascio della versione finale di un prodotto software. Se ne distinguono diversi tipi, elencati di seguito:

- Manutenzione correttiva: ha come scopo la correzione di errori, bug, inesattezze, inefficienze, etc.;
- Manutenzione adattiva: ha come scopo l'adattamento a diverse tecnologie, ambiti e contesti;
- Manutenzione evolutiva: ha come scopo l'adattamento a nuove tecnologie, ambiti, contesti, implementando nuove funzionalità e caratteristiche.

Miglioramento continuo

Principio attorno al quale organizzare i processi per ottenere un miglioramento continuo, prevede 4 macro-fasi: *Plan (keep track of what you're going to do)*, *Do (as planned)*, *Check*, *Act (keep what works, throw what doesn't)*.

Milestone

Letteralmente pietra miliare, rappresenta un traguardo (anche intermedio) di particolare rilevanza per l'andamento del progetto.

Minimizzazione

Riduzione ad icona.

Modello di ciclo di vita

Il ciclo di vita di un software non è univocamente determinato, ma ne vengono individuati diversi modelli possibili. La scelta del modello dipende da 3 macro-fattori: cosa vuole il committente, dipendenza da terze parti, livello di coinvolgimento del committente nell'accertamento dello stato di avanzamento.

Sequenziale (a cascata)

Ha per principio cardine la ripetibilità dei processi. Il ciclo di vita sequenziale è lineare, le fasi si susseguono e la direzione ammessa è una sola. Il modello fa forte uso di documentazione il che rende il sistema organizzato e tracciabile, prevede *pre* e *post* per ogni fase e associa ad ogni fase date di inizio e di fine. ISO 12207 definisce così le fasi del ciclo di vita sequenziale: analisi, progettazione (intesa come *design*), realizzazione, manutenzione.

Incrementale

Prevede un approccio che fa uso di incrementi, ha come vantaggi:

- Il valore aggiunto di ogni incremento;
- La riduzione del rischio di fallimento portata da ogni incremento;
- L'uso di *enablers* che facilitano il lavoro.

Evolutivo

Fa uso massiccio della fase di manutenzione viene fatta ad ogni versione rilevante del prodotto.

A componenti

Si sviluppa sull'idea di riutilizzare componenti. Prevede quindi tecniche di adattamento delle componenti e dei requisiti finalizzate al riuso del software.

A spirale

Modello di ciclo di vita utilizzato quando il progetto è innovativo e non esistono *best practice* applicabili per lo sviluppo del progetto.

Agili

Metodi che si rifanno a 4 principi fondamentali:

- Focus sugli *stakeholder* e le loro interazioni, piuttosto che su processi e strumenti;
- Focus sul software piuttosto che sulla documentazione;
- Rapporto collaborativo piuttosto che contrattuale con il cliente;
- Risposta veloce anche se non pianificata al cambiamento.

Tali metodi offrono degli svantaggi poiché la documentazione è cruciale nella fase di manutenzione del software e l'assenza di pianificazione di fronte ai cambiamenti comporta dei rischi.

Scrum

Scrum ha come focus fondamentale le *user-stories*: una visione di ciò che vuole il cliente. Da esse si ricava un *product backlog*, insieme di attività/feature da svolgere/realizzare. L'attività

di *Sprint Backlog* è un passo iterativo che impone controlli frequenti (*Daily Scrum*) sulla correttezza delle azioni intraprese, tramite *Sprint Review* e *Retrospective*.

Modulo

Si tratta di una componente o parte del software, un modulo:

- Incapsula dati e comportamenti in maniera coerente e coesa;
- Ha un'interfaccia che ne permette l'uso;
- E' pensato in modo da favorire la pratica della modularità.

Monitoraggio dei rischi

Attività che ha lo scopo di anticipare l'insorgere di eventi di rischio per mitigarne gli effetti.

O

Obiettivo SMART

- **Specific:** formalmente definito;
- **Measurable:** il cui stato di ottenimento è misurabile;
- **Achievable:** raggiungibile;
- **Realistic:** pratico;
- **Time-Bound:** limitato da un vincolo temporale.

Open source

Tipo di prodotto software il cui codice sorgente viene rilasciato al pubblico mediante una licenza che ne permette a chiunque lo studio, la modifica e la distribuzione per qualsiasi scopo.

P

Panel

Un *panel* è una rappresentazione visiva di dati elaborati da un sistema mostrati sotto-forma di grafici.

PDCA

Acronimo di *Plan, Do, Check, Act*, anche noto come ciclo di Deming. Si tratta di un ciclo di miglioramento continuo dei processi e dei prodotti, suddiviso in quattro fasi, che sono appunto *Plan, Do, Check, Act*.

Pianificazione delle attività

Pianificare le attività è uno dei compiti del project manager. La pianificazione include l'allineamento delle attività su un asse temporale e l'assegnazione delle attività a delle persone. Il project manager si avvale di diversi strumenti per pianificare le attività:

- **Diagrammi di Gantt:** Rappresentano durata prevista vs durata effettiva delle attività;
- **Diagrammi PERT:** *Programme Evaluation and Review Technique*, tiene conto delle dipendenze tra attività e mostra lo slack;
- **WBS:** *Work Breakdown Structure*, divisione delle attività fino a raggiungere il compito minore assegnabile ad una persona. Diagramma che rappresenta l'assegnazione di tali attività alle persone nel tempo, rispettando e rappresentando dipendenze e vincoli temporali.

Piano di progetto

Include:

- Introduzione (scopo, struttura);
- Organizzazione del progetto;
- Analisi dei rischi;
- Risorse disponibili (tempo, persone);
- Suddivisione del lavoro;
- Calendario delle attività;
- Meccanismi di controllo e rendicontazione.

Plug-in

Un *plug-in* in campo informatico è un programma non autonomo che interagisce con un altro programma per ampliare o estendere le sue funzionalità originarie. Ad esempio, un *plug-in* per un software di grafica permette l'utilizzo di nuove funzioni non presenti nel software principale.

PNG

Formato *lossless* per memorizzare immagini.

Processo

Insieme di attività correlate (includono tutto ciò che è attinente) e coese (sono tutte necessarie) che trasformano ingressi (bisogni) in uscite (prodotti) secondo regole date dal controllo processo a seguito di decisioni prese sulla base di misurazioni delle risorse consumate dal processo. I processi si differenziano in base alla specificità dell'ambito di applicazione, possono essere standard (riferimento base generico), definito (standard adattato alle esigenze aziendali), di progetto (definito adattato al progetto).

Processo software

Insieme di attività che devono essere svolte per far avanzare un prodotto software nel suo ciclo di vita (nell'ambito SWE, vedi "Processo" per una voce più generale).

Processo specializzato per progetto

Prevedono una fase di pianificazione e una di definizione, attenzione nella conduzione e analisi critica del funzionamento del processo.

Prodotto software

Una "entità" software progettata per essere rilasciata ad un cliente. Prende forme diverse in base al soggetto richiedente:

- **Commessa:** si tratta di un prodotto software le cui specifiche ed obiettivi sono indicati da un committente;
- **Pacchetto:** si tratta di un prodotto software le cui specifiche ed obiettivi sono indicati per la replicazione in o per altri software;
- **Componente:** si tratta di un prodotto software le cui specifiche ed obiettivi sono indicati per la composizione con altri software;
- **Servizio:** si tratta di un prodotto software le cui specifiche ed obiettivi sono determinati per la risoluzione di un problema.

Prodotti documentali

- Capitolato d'appalto;
- Studio di fattibilità;
- Analisi dei requisiti;

Produttività

Una misurazione di efficienza, rapporto tra quantità di prodotto realizzato e risorse consumate.

Progettazione software

Insieme di attività che precede la realizzazione il cui scopo è quello definire l'architettura del software, con l'obiettivo di sviluppare un prodotto corretto per costruzione piuttosto che per correzione.

Progetto

Insieme di attività e compiti atti a raggiungere obiettivi SMART, che hanno una data di inizio ed una di fine e che consumano un pool di risorse limitate.

Project manager

Individuo che ha il compito di gestione del progetto. Ciò prevede:

- Istanziare i processi nel progetto (quelli standard aziendali e quelli istanziati dai processi aziendali);
- Stimare costi e risorse necessarie;
- Pianificare attività (organizzarle nel tempo e a assegnandole);
- Controllare le attività e verificare i risultati.

Proponente

Parte terza, per esempio un'azienda, che propone un capitolato.

Prototipo

Esemplare di prova del prodotto. Serve a provare e scegliere soluzioni; può avere carattere usa e getta (in metodi iterativi) oppure essere la base per incrementi successivi (in metodi incrementali). Un prototipo di tipo usa e getta comporta dei costi che producono poco valore aggiunto.

Q

Qualità

Misura in cui un prodotto software soddisfa un certo numero di aspettative rispetto sia al suo funzionamento sia alla sua struttura interna.

Qualità del prodotto

Caratteristiche di un prodotto di qualità:

- **Sufficienza:** capacità di soddisfare tutti i requisiti;
- **Comprensibilità:** facilità di utilizzo e comprensione del funzionamento da parte degli *stakeholder*;
- **Modularità:** suddivisione in parti chiaramente distinte e sconnesse, principio fondamentale dell' *information hiding*;
- **Robustezza:** agli input di utilizzo;
- **Sicurezza:** affidabilità in caso di malfunzionamento (hardware o software), resistenza alla intrusioni;
- **Flessibilità:** misura della facilità con la quale il sistema può essere modificato o adattato per soddisfare nuovi requisiti;
- **Riusabilità:** misura della facilità di riutilizzo in altri contesti dei moduli che formano il prodotto;
- **Disponibilità:** più inconveniente il tempo di *downtime* in caso di guasti o aggiornamenti, meno disponibile risulta il prodotto;
- **Efficienza.**

Qualità dei processi

Una buona qualità dei processi induce una buona qualità del prodotto di questi. Per assicurarsi che un processo sia di qualità, è necessario un meccanismo di controllo che effettui modifiche sul way of working ove ritenuto necessario sulla base di misurazioni sulla qualità dei processi. Per definire gli strumenti di valutazione della qualità dei processi dei fornitori in genere, non solo in ambito di sviluppo software, è stato messo a punto lo standard ISO 9000, in ambito specifico di ingegneria software si ha invece ISO 90003:2004.

Quantificabile

Misurabile.

R

Raintank

Raintank è una piattaforma di monitoraggio open source per raccogliere, archiviare e analizzare enormi quantità di dati diagnostici, da un'ampia varietà di fonti e in molti formati diversi.

Repository

Termine inglese che significa "deposito", "ripostiglio". In ambito informatico viene usato per indicare una struttura dati in cui vengono gestiti i metadati per un insieme di file o una struttura di cartelle. Esempi di metadati gestiti da una *repository* sono lo storico dei cambiamenti avvenuti, o l'insieme dei commit (modifiche).

Requisito

I requisiti sono le capacità che il sistema dovrà avere per svolgere le funzioni volute dal committente. È best practice definire requisiti:

- **Non ambigui:** interpretati univocamente da tutti;
- **Corretti;**
- **Completi;**
- **Variabili;**
- **Consistenti:** che non si contraddicono a vicenda o da sé;
- **Modificabili;**
- **Tracciabili;**
- **Ordinati:** tra obbligatori, desiderabili ed opzionali.

Rete Bayesiana

Una *rete Bayesiana* è un modello grafico probabilistico che rappresenta un insieme di variabili stocastiche con le loro dipendenze condizionali attraverso l'uso di un grafo. Per esempio, una *rete Bayesiana* potrebbe rappresentare la relazione probabilistica esistente tra dei sintomi e delle malattie. Dati i sintomi, la rete può essere usata per calcolare la probabilità della presenza di diverse malattie.

Rischio

I rischi di progetto sono: sforamento dei costi, o dei tempi, o risultati insoddisfacenti. Fonti di rischio principali sono: tecnologie usate, rapporti interpersonali, organizzazione del lavoro, requisiti e rapporti con *stakeholder*, tempi e costi. La gestione del rischio si fa tramite:

identificazione (nel progetto, prodotto, mercato), analisi (probabilità e conseguenze), pianificazione (come evitare e o mitigare gli effetti) e controllo (attenzione agli indicatori di rischio, raffinamento strategie).

Riuso

Riutilizzo del codice, può essere di due tipi: opportunistico (copia e incolla), caratterizzato da un basso costo e poco impatto, o sistematico, caratterizzato da un maggior costo ed maggior impatto.

Rule Declaration

All'interno del linguaggio CSS una rule declaration possiede la seguente struttura:

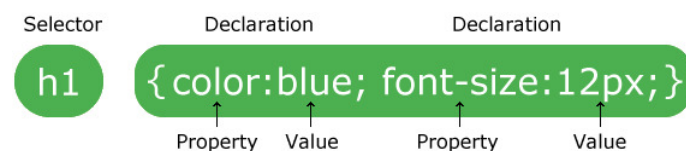


Figura 1: Struttura di una rule declaration
Immagine presa da w3schools.com

Dove:

- Il selettore punta all'elemento HTML che si vuole stilizzare;
- Il blocco di dichiarazioni contiene una o più dichiarazioni separate da un punto e virgola;
- Ogni dichiarazione è composta da due componenti separate dal carattere due punti:
 - il nome di una proprietà CSS;
 - un valore da associare ad essa.

Ruolo

Indica l'area di specializzazione. In un progetto informatico i ruoli principali sono:

- **Analista:** si occupa dell'analisi dei requisiti. In generale, individua e definisce il "problema" in termini formali, in modo che sia possibile verificare se la soluzione è tale;
- **Progettista:** si occupa del *design* del prodotto. In generale, si occupa della soluzione del "problema";
- **Programmatore:** Realizza il prodotto. Non ha libertà di scelta e segue le direttive del *Progettista*;
- **Verificatore:** testa la qualità del prodotto e comunica feedback;
- **Amministratore:** garantisce il funzionamento dell'apparato informatico aziendale.

Tutte queste figure professionali sono coordinate nell'ambito di un progetto da un project manager.

S

SASS

SASS è un'estensione del linguaggio CSS che permette di utilizzare variabili, di creare funzioni e di organizzare il foglio di stile in più file.

SASS Style Guide

Vedi Airbnb CSS.

Schedule Variance

Schedule Variance (SV) è una metrica che indica se le attività di progetto sono in linea rispetto alla loro schedulazione. Può assumere valori:

- **Positivi:** indicano che le attività procedono più velocemente rispetto a quanto pianificato;
- **Negativi:** indicano che le attività procedono più lentamente rispetto a quanto pianificato;
- **Zero:** indica che le attività rispettano la loro schedulazione.

Generalmente viene calcolata: $SV = EV - PV$, dove SV indica la *schedule variance*, EV indica l'*earned value*, ovvero il valore del lavoro effettuato e PV indica il *planned value*, ovvero il costo del lavoro pianificato.

Script

Il termine *script*, in informatica, designa un tipo particolare di programma, scritto in una particolare classe di linguaggi di programmazione, detti linguaggi di *scripting*.

Sistematico

Metodico, rigoroso.

Slack (tempo)

Margine di tempo tra la fine di un'attività e la scadenza di fine attività.

Se è positivo, l'attività è prevista terminare prima della scadenza, quindi vi è un margine per eventuali imprevisti. Se è vicino o 0, tale margine è piccolo o assente; si introducono quindi dei rischi. Se è negativo, la scadenza è passata e l'attività non è stata portata al termine.

Software Engineering

Disciplina il cui scopo è la realizzazione di prodotti software; nota anche con l'acronimo SWE. SWE si occupa dell'organizzazione e della gestione delle attività, compiti e interazioni di un

team di sviluppatori, le quali hanno come obiettivo per l'attività di sviluppo e il software risultante Efficacia ed Efficienza. SWE si occupa inoltre delle metodologie di cura del progetto per l'intero ciclo di vita del software. SWE prevede un approccio sistematico, disciplinato e quantificabile all'attività di sviluppo.

Sorgente Dati

Fonte di dati interpretabili per ottenere informazioni.

Stima dei costi

Influenzata dalle dimensioni del progetto, dalle esperienze pregresse, dalla familiarità con le tecnologie adottate, dalla produttività dell'ambiente di lavoro, dalla qualità attesa.

Un modello algoritmico che può aiutare in ciò è il CoCoMo.

Studio di fattibilità

Studio economico realizzato per determinare se è vantaggioso partecipare ad una gara d'appalto.

T

Task

Attività associata ad un segmento temporale.

Ticket

Un *ticket* è l'associazione tra una task ed un assegnatario. Un sistema di ticketing è una piattaforma per la gestione di task, ticket e assegnatari.

Tracciamento dei requisiti

Attività di monitoraggio dell'evoluzione e della scoperta dei requisiti che possono cambiare durante lo svolgersi dello sviluppo del progetto. Tale attività viene svolta tramite apparati informatici appositi, le informazioni importanti sui requisiti di cui viene tenuta traccia sono: *status*, *origin*, assegnatari.

TravisCI

TravisCI è un servizio distribuito per la gestione della Continuous Integration all'interno di repository *GitHub*.

Trender

Trender è un sistema che permette la gestione e il tracciamento di requisiti, casi d'uso, attori, verbali, package, classi e loro metodi.

TSLint

Strumento software per l'analisi statica del codice. Verifica che il codice rispetti un set di regole (estendibile) pensate per garantire conformità stilistica, evitare errori comuni, facilitare la manutenibilità, etc.. *TSLint* è integrabile in svariati *IDE* e nella piattaforma TravisCI.

TypeScript

Estensione di javascript incentrata nell'aumentarne la scalabilità. Sviluppata e mantenuta da Microsoft. Tra le feature di interesse vi è il *typing* statico e la possibilità di sviluppare applicazioni sia *client-side* che *server-side*.

V

Validare

Attività di confronto tra i risultati ottenuti e quelli aspettati. Quest'attività si svolge al termine del progetto.

Verificare

Attività di verifica di presenza di errori e di rispetto del way of working. La verifica si manifesta tangibilmente sotto forma di test, effettuati sui moduli che compongono il prodotto software. Le forme di verifica sono 2: analisi dinamica e analisi statica. L'analisi dinamica prevede l'esecuzione del software mentre l'analisi statica prevede un'analisi del codice sorgente senza eseguirlo.

Verificatore di requisiti

Individuo che ha il compito di verificare i requisiti, utilizza le tecniche descritte in analisi statica.

Versionamento

Organizzazione e gestione delle versioni del software prodotte nel corso dello sviluppo.

Versione

Vedi baseline.

W

Way of working

Insieme di regole, pratiche, che caratterizzano l'attività di sviluppo. Il way of working è pensato per agevolare lo sviluppo ed evitare sprechi di risorse ed errori.

Webhook

Tecnologia nell'ambito del *web development* che permette di modificare il comportamento di una pagina web tramite *callbacks* personalizzate.

Workflow

Termine inglese che significa "flusso di lavoro". Indica la gestione dei processi lavorativi.

X

XML

Si tratta di un meta linguaggio che permette di definire linguaggio di markup.

Z

Zero-latency

Termine inglese che significa "senza ritardo". Riferito al modo con cui si approccia il lavoro, prevede l'inizio delle attività di sviluppo non appena possibile. Permette di avere un margine di tempo tra il termine previsto del progetto e le scadenze imposte.

Zero-laxity

Termine inglese che significa "senza margine". Riferito al modo con cui si approccia il lavoro, prevede l'inizio delle attività di sviluppo il più tardi possibile in modo tale che non sia presente alcun margine tra il termine previsto del progetto e le scadenze imposte.