Proprietà SW sicuro

H1 # Processo e prodotto

H3 ### Processo

H2

> **Processo**: come realizziamo il prodotto

Esistono diversi *modelli di processo*, in cui le fasi di sviluppo sono gestite e organizzate in modo diverso.

Qualità del processo

Esse riguardano i metodi utilizzati per lo sviluppo del SW. Vediamone alcune.

H4

Produttività

> Essa misura l'efficienza del processo di produzione del SW in termini di *velocità di consegna* del prodotto.

H6

Tempestività

> Essa misura la capacità del processo di produzione del SW di *valutare e rispettare i tempi di consegna* del prodotto

H6

Trasparenza

> Essa misura la capacità del processo di produzione del SW di *capire il suo stato attuale e tutti i suoi passi*

H6

H3

Prodotto (SW)

> **Prodotto**: cosa realizziamo (software)

Caratteristiche

H4

- Intangibile
- Malleabile
- Human intensive (non comporta alcun processo manifatturiero tradizionale)

Qualità del software

Le qualità sono classificate in due macro categorie:

H4

- Qualità **esterne** (***black-box view***)
- Qualità **interne** (***white-box view***)

Tali categorie sono connesse, in quanto non è possibile ottenere le qualità esterne se il SW non gode già delle qualità interne.

Qualità esterne

> Sono quelle percepite da un osservatore esterno.

H5

Esse riguardano essenzialmente le **funzionalità** del SW. Vediamone alcune.

Correttezza (o funzionalità)

> Un SW è **corretto** se rispetta le specifiche *funzionali* del progetto (assumendo che delle specifiche esistano!)

H6

È una proprietà matematica che stabilisce l'**equivalenza** tra il SW e la sua specifica. Ciò significa che la *correttezza* di un SW dipende esclusivamente dall'aderenza alle specifiche, le quali possono essere definite anche in modo errato. Specifiche sbagliate possono dipendere da requisiti incorretti o da errori nella conoscenza del dominio dell'applicazione.
Se le specifiche sono espresse *formalmente*, la correttezza è provabile *formalmente* (***verifica***). Altrimenti, le prove possono avvenire attraverso

contro-esempi (***testing***).
Il ***testing*** NON garantisce la correttezza, mentre la ***verifica*** sì.

Affidabilità (o *reliability*)

> Un SW è **affidabile** se si comporta "come previsto"

H6

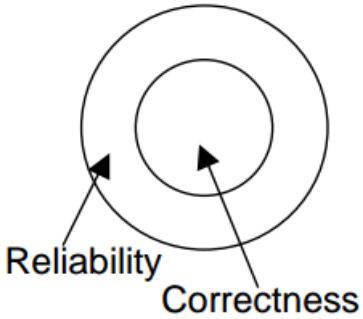
Matematicamente, può essere valutata come:

> La probabilità di assenza di fallimenti in un dato intervallo di tempo

La **correttezza** implica l'**affidabilità**, ma NON vale il contrario.

$$\text{correttezza} \implies \text{affidabilita}$$

$$\neg(\text{affidabilita} \implies \text{correttezza})$$



Il ***testing*** può provare che un SW è **affidabile**.

Efficienza

> Un SW è **efficiente** se usa intelligentemente le risorse di calcolo

H6

Può essere valutata attraverso *analisi di complessità* o la simulazione di scenari critici.

Usabilità

> L'**usabilità** di un SW è la facilità d'uso da parte dell'utente.

H6

È una qualità difficile da valutare, in quanto spesso è soggettiva. Inoltre implica il concetto di *expected user*.

Portabilità

> Un SW è **portabile** se può funzionare su più piattaforme

H6

Interoperabilità

> È l'abilità di un sistema di coesistere e cooperare con altri sistemi

H6

Robustezza

> Un SW è **robusto** se si comporta in modo ragionevole anche in circostanze non previste dalle specifiche(input errati, guasti HW...)

H6

La valutazione di correttezza e affidabilità sono basate sulle specifiche, mentre la robustezza riguarda i casi non trattati.

Qualità interne

Riusabilità

H5

> Un SW è **riusabile** se può essere usato, in tutto o in parte, per costruire nuovi sistemi

H6

Verificabilità

> È la possibilità di dimostrare a posteriori la *correttezza* o altre caratteristiche del SW

H6

La dimostrazione, come abbiamo visto, può avvenire attraverso:

- Verifica formale
- Testing

Facilità di manutenzione

> È la facilità nel realizzare *adattamenti, evoluzioni, correzioni* del SW

H6

Un SW è facile da manutenere se:

- è strutturato in modo da *facilitare la ricerca degli* ***errori*** (**modifiche correttive**)
- la sua struttura permette di *aggiungere nuove funzionalità* al sistema (**modifiche perfettive**)
- la sua struttura permette di *adattarlo ai cambiamenti del dominio applicativo* (**modifiche adattive**)

H3

Sicurezza del SW

All'interno di **sicurezza** vi sono due concetti più precisi e specifici:

- safety:
  - Il sistema non causa danno *verso* l'esterno
  - cose "cattive" non accadono *dal* sistema
- security:
  - Il sistema non subisce danni *dall'* esterno
  - cose "cattive" non accadono *sul* sistema

In altre parole, un sistema è *safe* se la sua esecuzione non causa danni all'ambiente esterno (qualunque esso sia), mentre un sistema è *secure* quando l'ambiente esterno non riesce a danneggiarlo.

Noi tratteremo la sicurezza in termini di ***security***.

Tale proprietà NON è *assoluta*, ma *relativa* al contesto dell'applicazione, cioè alla ***policy di sicurezza*** definita per tale applicazione, che può essere introdotto intuitivamente come l'insieme di risposte alla domanda *"da chi/cosa il sistema deve essere sicuro?"*

Proprietà del SW sicuro

Un SW sicuro deve godere di ulteriori qualità rispetto a quelle comuni a tutti i SW che abbiamo visto prima.

H4

Prevenzione

> La capacità del SW di *anticipare* possibili *attacchi*

H6

Gli attacchi possono essere a livello di:

- progettazione
- implementazione
- d'uso

Per valutare tale proprietà nella fase di sviluppo viene svolta l'attività chiamata **analisi del rischio**, che mi permette di individuare i rischi a cui può essere soggetto il sistema.

Tracciabilità

> È il meccanismo che consente di stabilire in modo inequivocabile la *relazione causa-effetto* tra elementi, processi, o eventi

H6

In altre parole, ho la possibilità di *tracciare* l'esecuzione del sistema per ripercorrere la concatenazione degli eventi. Ciò è utile per riparare a un attacco o malfunzionamento. Un esempio semplice sono i file di log.

Controllo (*auditing*)

> Per **auditing** si intende il *confronto* tra le attività svolte sul sistema con le politiche stabilite (*policy*) al fine di determinare la loro conformità

H6

Monitoraggio

> Il **monitoraggio** è *auditing in tempo reale*

H6

Voglio controllare la conformità fra il sistema e la policy durante l'esecuzione del sistema stesso. Ciò si può ottenere in diversi modi, ad esempio tramite *asserzioni* (`assert` in Java, ad es.), che sono controlli logici sullo stato dei valori in memoria. Esse vanno però ad inficiare sull'efficienza.

H6 **Privatezza**

> È il diritto di un individuo di sapere *se, come, quando* e *a chi* un'informazione che lo riguardo può essere rilasciata

H6 **Confidenzialità**

> È la proprietà che assicura che certi servizi e informazioni del sistema siano accessibili solo ad utenti autorizzati

H6 **Sicurezza a diversi livelli**

> Possibilità del sistema di avere *diversi* ***livelli di sicurezza***

H6 **Anonimato**

> La proprietà di *mantenere segreta l'**origine** di certi dati*

H6 **Autenticazione**

> La proprietà di *conoscere l'**identità** di chi accede al servizio*

H6 **Integrità**

> La proprietà per cui le informazioni e le risorse del sistema siano *integre*, ovvero che *non vengano modificate o cancellate in modo non autorizzato o improprio*