

# L'INFORMATICA È UNIVERSALE?

---

S. BELLINELLI - 05.05.2022



# QUANTO DOBBIAMO CORRERE ?

---

«Tutti ti dicevano che perdi perché non corri abbastanza, un giorno arriva lui e ti spiega che perdi perché corri troppo»  
(Pep Guardiola, su Johan Cruyff)



# QUANTO DOBBIAMO CORRERE? (2)

---

- Non fare tutto, ma ciò che serve
- Rilasciare il software solo dopo il «ciclo completo»
- Favorire piccole messe in produzione regolari
  - Evitano la sensazione di abbandono
  - Riducono il rischio di regression

# TRASPARENZA

---

«Le bugie hanno le gambe corte»  
(Tua mamma)



# TRASPARENZA (2)

---

- Valutare bene i tempi di consegna
- Non promettere più di quanto siamo in grado di fare
- Non nascondere i difetti, ma discuterli e affrontarli

# IL GHIACCIO

---

«Quando si pattina su ghiaccio sottile,  
la salvezza sta nella velocità»  
(Ralph Waldo Emerson)



# IL GHIACCIO (2)

---

Due tipi di velocità:

- Velocità nel trovare l'errore

La presenza di test, velocizza la ricerca

- Velocità nel correggerlo

Un codice «pulito» permette di effettuare modifiche in minor tempo

# L'ARTE DELLA GUERRA

---

«Se conosci il nemico e conosci te stesso,  
nemmeno in cento battaglie ti troverai in pericolo»  
(Sun Tzu, L'arte della guerra)



# L'ARTE DELLA GUERRA (2)

---

- Ottima conoscenza del business
- Buona padronanza dei mezzi tecnici
- Definizione degli acceptance test con il business

# L'ARTE DELLA GUERRA (3)

---

«Nell'operazione militare vittoriosa,  
prima ci si assicura la vittoria, poi si dà battaglia»  
(Sun Tzu, L'arte della guerra)



# L'ARTE DELLA GUERRA (4)

---

Pianificazione della vittoria:

- Definizione degli acceptance test

Sviluppo della battaglia:

- Sviluppo del codice e dei test unitari

# IL VECCHIO WEST

---

«La velocità è cosa buona,  
ma l'accuratezza è tutto»  
(Wyatt Earp)



# IL VECCHIO WEST (2)

---

- Studio degli use case con il business
- Selezionare accuratamente le storie
- Ordinare in base alla priorità

# APERTURA ALLE MODIFICHE

---

«You miss 100% of the shots you DON'T take»

«A good hockey player plays where the puck is.  
A great one, plays where the puck is going to be»

(The Great One)



# APERTURA ALLE MODIFICHE (2)

---

- Rendere il codice facilmente modificabile
- Rendere il codice autoesplicativo
- Mantenere aggiornati i test

# NON È MAI TROPPO TARDI

---

«Se stai pensando di prendere una mano,  
vuol dire che è tardi»

(Antico proverbio)



# NON È MAI TROPPO TARDI (2)

---

- Strutturare il codice in modo da aprirci ai cambiamenti
- Test non legati ai dettagli implementativi
- Evitare la «pigrizia a breve termine»

# SAGGEZZA ORIENTALE

---

«近朱者赤， 近墨者黑»

«Chi sta vicino al vermiglio diventa rosso,  
chi sta vicino all'inchiostro, diventa nero»

(Proverbio cinese)



SAGGEZZA ORIENTALE (2)

---

**BEHAVIOUR ATTRACTORS**



# CONCLUSIONI

---

- Concetti comuni non limitati all'informatica
- Soluzioni semplici, ma sforzo continuo
- Con il passare del tempo lo sforzo diminuisce
- Tutto il team deve collaborare al mantenimento dello stato ottimale

# Q & A

---



# THANKS FOR WATCHING

---

Stefano Bellinelli, Software developer at EOC  
stefano.bellinelli@eoc.ch