

Work to do (14 Novembre)

Dato un DDR-Unibo-robot bisogna costruire un sistema che:

- Consenta all'utente di inviare comandi (w, s, a, d, h)
- Consenta all'utente di conoscere lo stato di moto del robot
- Permette di attivare/deattivare un'attività di 'esplorazione' in cui il robot:
 - Si muove in modo autonomo in un territorio piano
 - Evita eventuali ostacoli
 - Si ferma ad un comando di STOP inviato dall'utente
 - Si ferma nel caso in un segnale di allarme
 - Cerca di creare una mappa de territorio in cui si muove

Metodologia di sviluppo

È necessario lavorare secondo il criterio degli Sprint di Scrum in modo da vedersi ad ogni sprint per discutere della deliverable (lavoro prodotto).

Capire come si fa in termini concreti lo sviluppo incrementale basato sulla continuous integration.

- Quando inizio ad occuparmi del Testing del sistema?
- Cosa avrò in mano quando ho finito la fase di Analisi dei Requisiti?
- Cosa avrò in mano quando ho finito la fase di Analisi del Problema?
- Ammettendo di seguire un approccio incrementale evolutivo sarò in grado, e fino a che punto, di implementare dei sistemi che evolvono in maniera monotona crescente? Andrò sempre dritto verso la meta oppure devo mettere in conto che avrò dei ripensamenti durante la fase di feedback?
- Ma che cosa intende realmente Scrum quando parla di Sprint Review?
- Cosa intende realmente Scrum quando parla di Sprint Retrospective?

Lo scopo è costruire un prodotto software di qualità in maniera consapevole (quindi saper dire in ogni momento cosa sto facendo e perché).

- Mi interessa che qualcosa sopravviva alla fine di questo progetto o no? Sono interessato che alla fine del progetto rimanga qualcosa di riutilizzabile?

Uno dei punti chiave della Agile Programming è: non fare niente di più di quello che è richiesto dai requisiti, ciò nonostante....

Man mano che vado avanti, come impresa, in un certo dominio applicativo, come posso accumulare know-how riutilizzabile?

- Se fossi un responsabile aziendale, come organizzo il team? Quali competenze devo ricercare nel team di sviluppo dato un certo progetto?

Grande ribaltamento: passare dal bottom-up al top-down.

Più il problema è significativo/complesso più diventa essenziale e strategico il ruolo dell'Analisi (sia dei requisiti sia del problema) come premessa ad una corretta progettazione. Il resto dovrebbe essere implementazione.

Affrontare il problema al giusto livello concettuale in fase di analisi.

- Chi decide il giusto livello concettuale con cui aggredire le fasi iniziali di analisi?

Strategia: devo pensare da Analista e non da Progettista.