**📌 Prompt 1 – Generazione WBS (Work Breakdown Structure)**

**Ho a disposizione le specifiche funzionali e tecniche preliminari del requisito “Nuovo report dettaglio consegne”.**

**Usa il repository storico e proponi una WBS preliminare con attività granulari.**

**Formato output richiesto:**

**- ID attività**

**- Descrizione attività**

**- Deliverable atteso**

**- Ruolo responsabile (es. sviluppatore backend, frontend, QA, architetto)**

**- Predecessori (se presenti)**

**👉 Output atteso (esempio):**

* **A1 – Analisi dettagliata requisito → Documento analisi → Business Analyst**
* **A2 – Estensione schema DB consegne → Nuova tabella consegne\_firma → DBA → Predecessore: A1**
* **A3 – Creazione API “GET /delivery/signature” → Endpoint API REST → Dev Backend → Predecessore: A2**
* **A4 – Modifica UI portale → Nuovo pulsante export Excel → Dev Frontend → Predecessore: A3**
* **A5 – Test integrazione end-to-end → Report test → QA → Predecessore: A4**

**📌 Prompt 2 – Stima effort preliminare**

**Sulla base della WBS generata e confrontando con progetti simili presenti nel repository:**

**- Assegna a ciascuna attività una stima effort preliminare (giorni/uomo).**

**- Indica anche un livello di confidenza (Alta, Media, Bassa) a seconda della copertura di casi precedenti nel repository.**

**- Restituisci i risultati in formato tabella.**

**👉 Output atteso (esempio):**

| **Attività** | **Effort stimato (gg/uomo)** | **Confidenza** |
| --- | --- | --- |
| **A1 Analisi requisito** | **2** | **Alta** |
| **A2 Estensione schema DB** | **3** | **Media** |
| **A3 Creazione API** | **5** | **Alta** |
| **A4 Modifica UI portale** | **4** | **Media** |
| **A5 Test integrazione** | **3** | **Bassa** |

**📌 Prompt 3 – Identificazione dipendenze**

**Analizza la WBS e:**

**- Costruisci una mappa delle dipendenze tra attività.**

**- Identifica quali attività possono essere eseguite in parallelo e quali sono sequenziali.**

**- Evidenzia i colli di bottiglia critici.**

**Restituisci sia una lista testuale sia un diagramma logico (ASCII o PlantUML).**

**👉 Output atteso (esempio testuale):**

* **A1 → A2 → A3 → A4 → A5**
* **Attività parallele: A2 (DB) e A4 (UI) possono procedere in parte in parallelo se definita l’API mock.**
* **Colli di bottiglia: A3 (API) è critico perché blocca sia l’UI che i test end-to-end.**

**📌 Prompt 4 – Generazione Gantt**

**Usa la WBS, effort stimato e dipendenze per generare un Gantt preliminare.**

**- Timeline su scala settimane.**

**- Evidenzia milestone principali.**

**- Marca i colli di bottiglia critici.**

**Restituisci output in formato tabellare (inizio/fine attività) e, se possibile, in ASCII Gantt semplificato.**

**👉 Output atteso (esempio tabellare):**

| **Attività** | **Inizio** | **Fine** | **Milestone** |
| --- | --- | --- | --- |
| **A1 Analisi requisito** | **05/09** | **06/09** | **–** |
| **A2 Estensione schema DB** | **07/09** | **09/09** | **–** |
| **A3 Creazione API** | **10/09** | **16/09** | **Milestone API pronta** |
| **A4 Modifica UI portale** | **13/09** | **18/09** | **–** |
| **A5 Test integrazione** | **19/09** | **21/09** | **Milestone test completati** |

**📌 Prompt 5 – Identificazione rischi e risorse**

**Analizza la WBS e proponi:**

**- I principali rischi tecnici o organizzativi.**

**- I colli di bottiglia che potrebbero causare ritardi.**

**- Il registro delle risorse coinvolte (solo ruoli, non nomi).**

**Formato: tabella con Rischio → Probabilità → Impatto → Mitigazione.**