**Piano di Rifattorizzazione - Struttura e Delegazioni**

book\_builder.py --> Moduli Rifattorizzati

──────────────────────────────────────────────────────────────────────────────────────

send\_to\_genspark() --> ai\_interfaces/genspark\_driver.py

get\_last\_response() --> ai\_interfaces/genspark\_driver.py

check\_for\_generation\_error() --> ai\_interfaces/genspark\_driver.py

handle\_consecutive\_errors() --> ai\_interfaces/genspark\_driver.py

get\_clean\_input\_box() --> ai\_interfaces/interaction\_utils.py

safe\_text\_input() --> ai\_interfaces/interaction\_utils.py

click\_send\_button() --> ai\_interfaces/interaction\_utils.py

wait\_for\_stable\_response() --> ai\_interfaces/interaction\_utils.py

split\_prompt() --> ai\_interfaces/interaction\_utils.py

handle\_context\_limit() --> ai\_interfaces/browser\_manager.py

reset\_context\_manual() --> ai\_interfaces/browser\_manager.py

take\_debug\_screenshot() --> ai\_interfaces/browser\_manager.py

create\_focused\_context() --> ai\_interfaces/context\_manager.py

send\_essential\_context() --> ai\_interfaces/context\_manager.py

check\_unresolved\_placeholders() --> ai\_interfaces/file\_text\_utils.py

format\_analysis\_results\_html() --> framework/formatters.py

process\_text() --> framework/formatters.py

process\_list\_html() --> framework/formatters.py

process\_patterns\_html() --> framework/formatters.py

process\_table\_html() --> framework/formatters.py

\_analyze\_market\_crisp() --> framework/analysis/market\_analysis.py

\_analyze\_market\_legacy() --> framework/analysis/market\_analysis.py

\_filter\_legacy\_prompt\_sections() --> framework/analysis/market\_analysis.py

update\_analysis\_status() --> ui/interface\_utils.py

show\_feedback() --> ui/interface\_utils.py

debug\_check\_components() --> ui/interface\_utils.py

\_generate\_book\_crisp() --> framework/book\_generator.py

\_generate\_book\_legacy() --> framework/book\_generator.py

generate\_book() --> framework/book\_generator.py

\_parse\_book\_index() --> framework/book\_generator.py

\_load\_chapter\_prompt() --> framework/book\_generator.py

\_handle\_missing\_placeholders() --> framework/book\_generator.py

\_save\_chapter() --> framework/book\_generator.py

execute\_with\_updates() --> framework/utils.py

**Processo per ogni funzione:**

1. **Copia** la funzione da book\_builder.py al modulo di destinazione
2. **Adatta** la funzione per funzionare come funzione indipendente:
   * Rimuovi i riferimenti a self
   * Aggiungi parametri espliciti per le dipendenze
   * Aggiorna gli import necessari
   * Mantieni la stessa logica di business
3. **Crea** una versione delegata in book\_builder.py:
   * Importa la funzione dal nuovo modulo
   * Chiama la funzione passando i parametri necessari dal contesto dell'oggetto