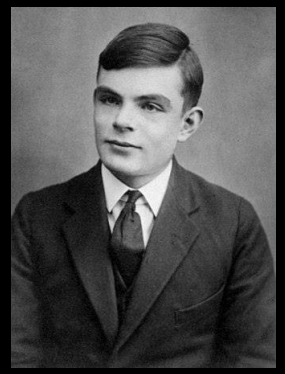
ENIGMA?
UN APPROCCIO TDD**?**

Se il business ti avesse chiesto un software per decifrare il codice Enigma

LA MACCHINA ENIGMA

- Dispositivo elettromeccanico per cifrare e decifrare messaggi
- Ampiamente utilizzato dalle forze armate tedesche durante il periodo nazista e della seconda guerra mondiale (le comunicazioni navali erano state decriptate dalla Gran Bretagna)
- Ampio utilizzo per facilità d'uso e presunta indecifrabilità
- Diverse versioni commerciali (open source) e diverse versioni sviluppate ad hoc per l'apparato militare tedesco

UN PO' DI STORIA

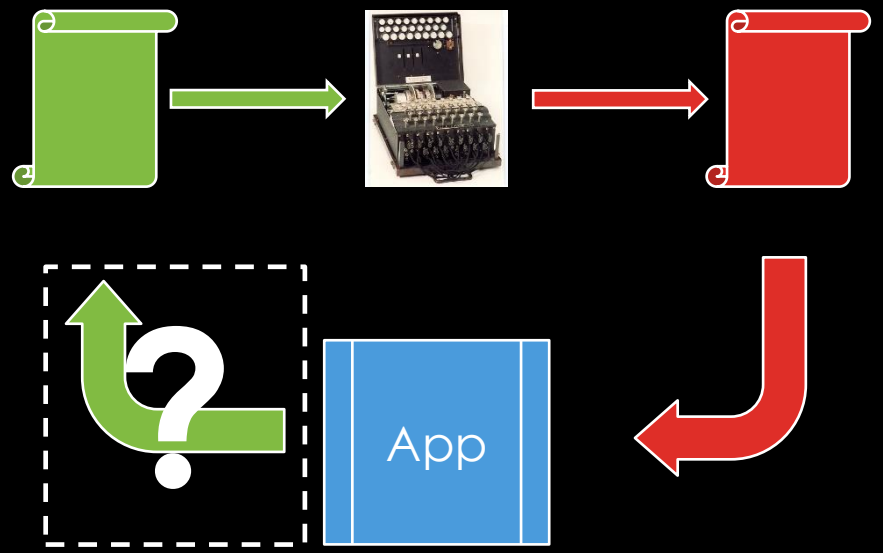


- 1918: Arthur Scherbius ottiene il brevetto per Enigma
- 1926: Marina militare tedesca ed Esercito iniziano ad usare Enigma per le comunicazioni
- 1933: l'intelligence polacco, guidato dal matematico Marian Rejewski, attraverso una macchina chiamata Bomba riesce a decifrare il codice Enigma
- 1939: Enigma-G (major release): i tedeschi introducono in Enigma un insieme di cinque rotori, dei quali sempre solo tre venivano usati, ma diversi ogni giorno: la Bomba polacca non poteva affrontare un tale incremento di complessità
- 1939: il progetto venne trasferito agli inglesi (Bletchley Park), con l'aiuto di grandi matematici come Alan Turing
- 1939: Bomba 'inglese' si focalizza su testo atteso (presunto) per scartare le combinazioni errate

FUNZIONAMENTO

- L'operatore riceveva il messaggio scritto (in chiaro)
- Premeva sulla tastiera effettiva una lettera del medesimo, sulla "tastiera luminosa" compariva la corrispondente lettera cifrata
- l'operatore registrava il messaggio criptato su un altro foglio
- il messaggio cifrato veniva consegnato al marconista che lo trasmetteva (via radio o via filo)
- Analogamente si procedeva in decrittazione: l'operatore di Enigma riceveva dal marconista o da chi per lui il messaggio cifrato, lo "batteva" sulla tastiera effettiva e le lettere in chiaro comparivano via via sulla "tastiera luminosa".

BUSINESS REQUEST

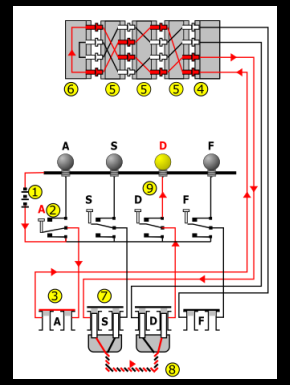


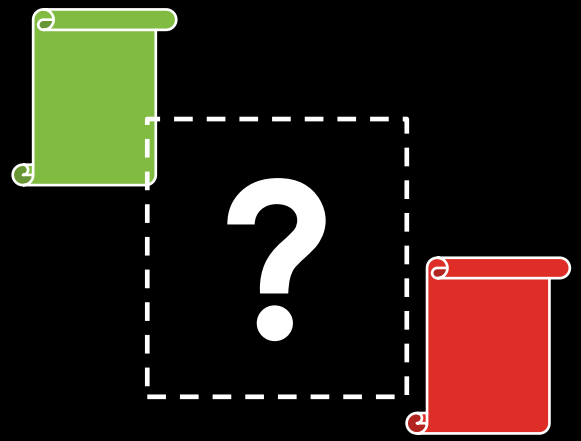
- Sviluppare un'applicazione che sia in grado di decifrare un messaggio cifrato con codice Enigma
- Non abbiamo una macchina Enigma
- Conosciamo il suo funzionamento
- Conosciamo alcuni vincoli (cosa non è possibile)
- Possiamo ipotizzare pratiche standard di chi trasmette

ANALISI SPECIFICHE

Specifiche di implementazione:

- La mappatura lettera per lettera si basa su 3 rotori attivi su 5 totali
- Ogni rotore ha 26 contatti disposti in maniera differente
- E' presente una board di configurazione tra le due tastiere





Specifiche di comportamento (cifratura):

- L'alfabeto tedesco è composto da 26 caratteri
- Ogni lettera viene cifrata con una lettera differente (A !=> A)
- La mappatura dei caratteri cambia ad ogni battitura (es. AA -> JE)
- La mappatura è simmetrica (AA -> JE, JE -> AA)
- Lo spazio viene definito sul messaggio in chiaro con una X
- I numeri vengono espressi in parola
- La configurazione cambia ogni giorno (reset condizione iniziale)

APPROCCIO TDD

Quali test? Qual è l'esito atteso?

- Iniziamo dai 'crib': riusciamo ad inferire la mappatura o il testo in chiaro basandoci sulla conoscenza parziale del testo in chiaro (errori)
  - Messaggio ritrasmesso il giorno dopo con codifica differente (se uno è già decodificato)
  - Messaggi standard ogni giorno ("The weather in the Bay of Biscay will be…")
  - Messaggi ritrasmessi al ri-emergere dei sotto-marini U-boat (rilevazione messaggi con gruppi semantici di lettere corrispondenti)
  - Frasi o parole ricorrenti ('heil Hitler', 'eins')

Cipher:	I	U	G	H	L	U	V	F	A	O	B	N	E	W	N	A	G	Z	W
Crib:	M	A	R	K	W	O	R	T	H	X	A	T	T	A	C	K	E	D	X

GESTIONE PROGETTO AGILE

In ogni momento:

- possono arrivare dettagli funzionali aggiuntivi sul comportamento di Enigma
- possono arrivare nuovi test (da nuovi crib) che falliscono con l'attuale implementazione -> necessaria triangulation sui test
- possono essere introdotti nuovi gradi di libertà su Enigma (es. plugboard o rotori variabili/aggiuntivi)

Storia

- Le prime 3 lettere non-cifrate indicavano come allineare i rotori (poi 6 cifre criptate configurazione per il messaggio)
- Maggio 1941: catturato un U-Boat con Enigma configurato, documenti implementativi, istruzioni d'uso, e codici di configurazione per 2 mesi
- Introduzione 4° rotore in Enigma

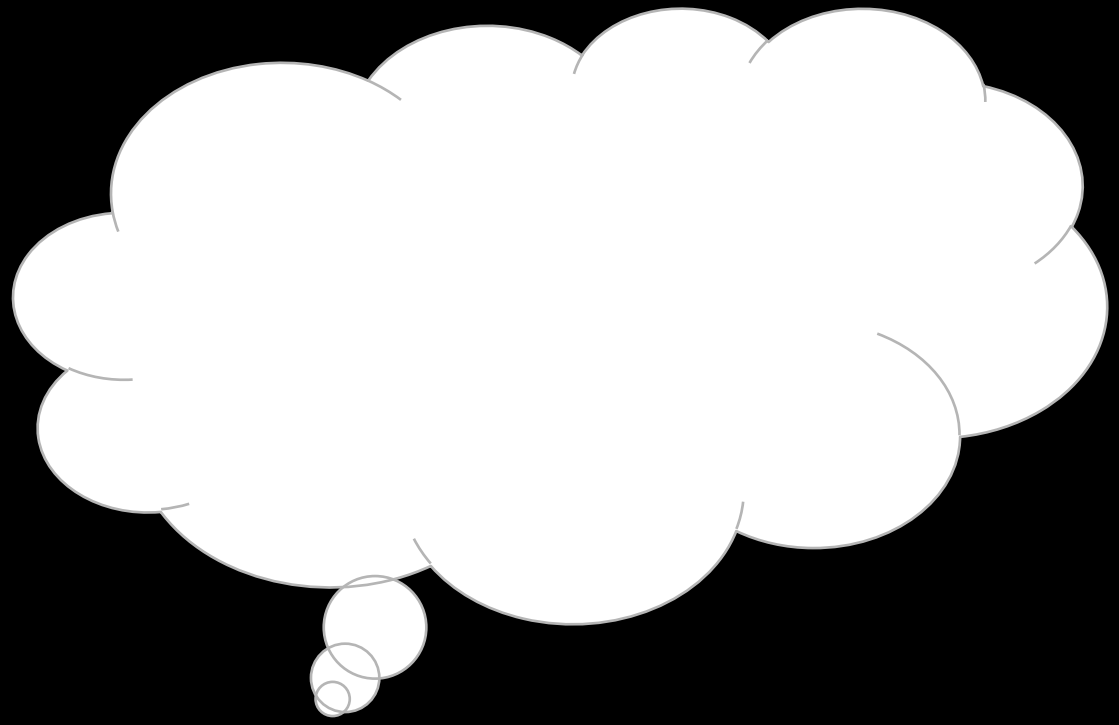
CONCLUSIONI E RIFLESSIONI

- L'esempio di Enigma in realtà non si configura come lo sviluppo di una semplice applicazione, ma richiede un algoritmo di 'ottimizzazione' dei parametri di configurazione
- Il processo non noto (Enigma) ha un comportamento deterministico, ma vista la complessità dei gradi di libertà ed il numero elevato di combinazioni il nostro simulatore deve effettuare una ricerca stocastica
- La ricerca ha un timeout fisiologico, l'algoritmo doveva essere veloce per poter interpretare i messaggi (spesso i messaggi riportavano le posizioni delle navi avversarie)

GRAZIE!

Limiti, esperienze o opinioni sull'approccio TDD in situazioni analoghe dove è impossibile testare tutte le combinazioni?

Domande



Curiosità

- X-Ray Imaging an Enigma Machine