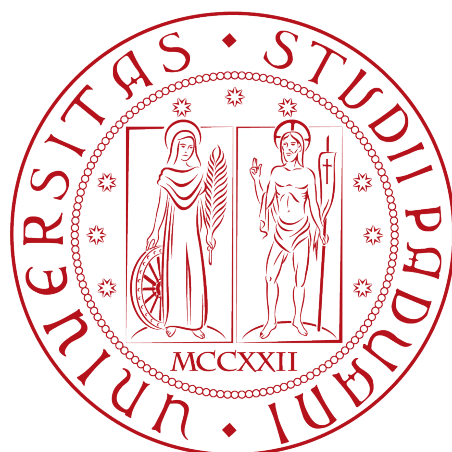


Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA “TULLIO LEVI-CIVITA”

CORSO DI LAUREA IN INFORMATICA



**Integrazione di processi PowerAutomate  
all'interno di applicazioni aziendali**

*Tesi di laurea*

*Relatore*

Prof. Tullio Vardanega

*Laureando*

Silvio Nardo

---

ANNO ACCADEMICO 2023-2024



# Sommario

Il presente documento descrive il lavoro svolto durante il periodo di stage, della durata di circa trecento ore, dal laureando Pinco Pallino presso l'azienda Azienda S.p.A. Gli obbiettivi da raggiungere erano molteplici.

In primo luogo era richiesto lo sviluppo di ... In secondo luogo era richiesta l'implementazione di un ... Tale framework permette di registrare gli eventi di un controllore programmabile, quali segnali applicati Terzo ed ultimo obbiettivo era l'integrazione ...

# Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
1.1	L'azienda . . . . .	1
1.2	L'idea . . . . .	1
1.3	Organizzazione del testo . . . . .	1
<b>2</b>	<b>Processi e metodologie</b>	<b>3</b>
2.1	Processo sviluppo prodotto . . . . .	3
<b>3</b>	<b>Descrizione dello stage</b>	<b>4</b>
3.1	Introduzione al progetto . . . . .	4
3.2	Analisi preventiva dei rischi . . . . .	4
3.3	Requisiti e obiettivi . . . . .	4
3.4	Pianificazione . . . . .	4
<b>4</b>	<b>Analisi dei requisiti</b>	<b>5</b>
4.1	Casi d'uso . . . . .	5
4.2	Tracciamento dei requisiti . . . . .	6
<b>5</b>	<b>Progettazione e codifica</b>	<b>8</b>
5.1	Tecnologie e strumenti . . . . .	8
5.2	Ciclo di vita del software . . . . .	8
5.3	Progettazione . . . . .	8
5.4	Design Pattern utilizzati . . . . .	8
5.5	Codifica . . . . .	8
<b>6</b>	<b>Verifica e validazione</b>	<b>9</b>
<b>7</b>	<b>Conclusioni</b>	<b>10</b>
7.1	Consuntivo finale . . . . .	10
7.2	Raggiungimento degli obiettivi . . . . .	10
7.3	Conoscenze acquisite . . . . .	10
7.4	Valutazione personale . . . . .	10
<b>A</b>	<b>Appendice A</b>	<b>11</b>
	<b>Acronimi e abbreviazioni</b>	<b>12</b>
	<b>Glossario</b>	<b>13</b>
	<b>Bibliografia</b>	<b>14</b>

# Elenco delle figure

4.1	Use Case - UC0: Scenario principale . . . . .	5
-----	---	---

# Elenco delle tabelle

4.1	Tabella del tracciamento dei requisiti funzionali . . . . .	7
4.2	Tabella del tracciamento dei requisiti qualitativi . . . . .	7
4.3	Tabella del tracciamento dei requisiti di vincolo . . . . .	7

# Capitolo 1

## Introduzione

Introduzione al contesto applicativo.

Esempio di utilizzo di un termine nel glossario  
[Application Program Interface \(API\)](#).

Esempio di citazione in linea  
*Manifesto Agile*. URL: <http://agilemanifesto.org/iso/it/>.

Esempio di citazione nel pie' di pagina  
citazione<sup>1</sup>

### 1.1 L'azienda

Descrizione dell'azienda.

### 1.2 L'idea

Introduzione all'idea dello stage.

### 1.3 Organizzazione del testo

[Il secondo capitolo](#) descrive ...

[Il terzo capitolo](#) approfondisce ...

[Il quarto capitolo](#) approfondisce ...

[Il quinto capitolo](#) approfondisce ...

[Il sesto capitolo](#) approfondisce ...

[Nel settimo capitolo](#) descrive ...

---

<sup>1</sup>Daniel T. Jones James P. Womack. *Lean Thinking, Second Editon*. Simon & Schuster, Inc., 2010.

Riguardo la stesura del testo, relativamente al documento sono state adottate le seguenti convenzioni tipografiche:

- gli acronimi, le abbreviazioni e i termini ambigui o di uso non comune menzionati vengono definiti nel glossario, situato alla fine del presente documento;
- per la prima occorrenza dei termini riportati nel glossario viene utilizzata la seguente nomenclatura: *parola*<sup>[g]</sup>;
- i termini in lingua straniera o facenti parti del gergo tecnico sono evidenziati con il carattere *corsivo*.

## Capitolo 2

# Processi e metodologie

*Brevissima introduzione al capitolo*

### 2.1 Processo sviluppo prodotto



## Capitolo 3

# Descrizione dello stage

*Breve introduzione al capitolo*

### 3.1 Introduzione al progetto

### 3.2 Analisi preventiva dei rischi

Durante la fase di analisi iniziale sono stati individuati alcuni possibili rischi a cui si potrà andare incontro. Si è quindi proceduto a elaborare delle possibili soluzioni per far fronte a tali rischi.

#### **1. Performance del simulatore hardware**

**Descrizione:** le performance del simulatore hardware e la comunicazione con questo potrebbero risultare lenti o non abbastanza buoni da causare il fallimento dei test.

**Soluzione:** coinvolgimento del responsabile a capo del progetto relativo il simulatore hardware.

### 3.3 Requisiti e obiettivi

### 3.4 Pianificazione

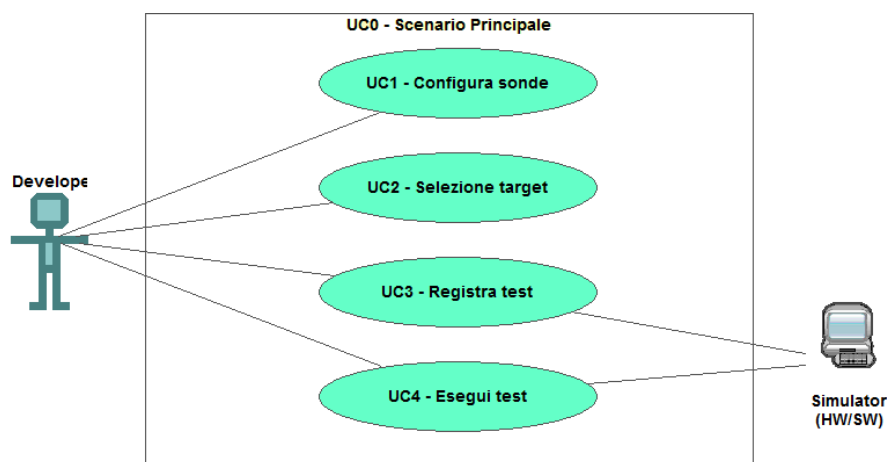
## Capitolo 4

# Analisi dei requisiti

*Breve introduzione al capitolo*

### 4.1 Casi d'uso

Per lo studio dei casi di utilizzo del prodotto sono stati creati dei diagrammi. I diagrammi dei casi d'uso (in inglese *Use Case Diagram*) sono diagrammi di tipo [Unified Modeling Language \(UML\)](#) dedicati alla descrizione delle funzioni o servizi offerti da un sistema, così come sono percepiti e utilizzati dagli attori che interagiscono col sistema stesso. Essendo il progetto finalizzato alla creazione di un tool per l'automazione di un processo, le interazioni da parte dell'utilizzatore devono essere ovviamente ridotte allo stretto necessario. Per questo motivo i diagrammi d'uso risultano semplici e in numero ridotto.



**Figura 4.1:** Use Case - UC0: Scenario principale

#### UC0: Scenario principale

**Attori Principali:** Sviluppatore applicativi.

**Precondizioni:** Lo sviluppatore è entrato nel plug-in di simulazione all'interno dell'IDE.

**Descrizione:** La finestra di simulazione mette a disposizione i comandi per configurare, registrare o eseguire un test.

**Postcondizioni:** Il sistema è pronto per permettere una nuova interazione.

## 4.2 Tracciamento dei requisiti

Da un'attenta analisi dei requisiti e degli use case effettuata sul progetto è stata stilata la tabella che traccia i requisiti in rapporto agli use case.

Sono stati individuati diversi tipi di requisiti e si è quindi fatto utilizzo di un codice identificativo per distinguerli.

Il codice dei requisiti è così strutturato  $R(F/Q/V)(N/D/O)$  dove:

R = requisito

F = funzionale

Q = qualitativo

V = di vincolo

N = obbligatorio (necessario)

D = desiderabile

Z = opzionale

Nelle tabelle [4.1](#), [4.2](#) e [4.3](#) sono riassunti i requisiti e il loro tracciamento con gli use case delineati in fase di analisi.

**Tabella 4.1:** Tabella del tracciamento dei requisiti funzionali

Requisito	Descrizione	Use Case
RFN-1	L'interfaccia permette di configurare il tipo di sonde del test	UC1

**Tabella 4.2:** Tabella del tracciamento dei requisiti qualitativi

Requisito	Descrizione	Use Case
RQD-1	Le prestazioni del simulatore hardware deve garantire la giusta esecuzione dei test e non la generazione di falsi negativi	-

**Tabella 4.3:** Tabella del tracciamento dei requisiti di vincolo

Requisito	Descrizione	Use Case
RVO-1	La libreria per l'esecuzione dei test automatici deve essere riutilizzabile	-

# Capitolo 5

## Progettazione e codifica

*Breve introduzione al capitolo*

### 5.1 Tecnologie e strumenti

Di seguito viene data una panoramica delle tecnologie e strumenti utilizzati.

#### **Tecnologia 1**

Descrizione Tecnologia 1.

#### **Tecnologia 2**

Descrizione Tecnologia 2

### 5.2 Ciclo di vita del software

### 5.3 Progettazione

#### **Namespace 1**

Descrizione namespace 1.

**Classe 1:** Descrizione classe 1

**Classe 2:** Descrizione classe 2

### 5.4 Design Pattern utilizzati

### 5.5 Codifica

## Capitolo 6

# Verifica e validazione

## Capitolo 7

# Conclusioni

7.1 Consuntivo finale

7.2 Raggiungimento degli obiettivi

7.3 Conoscenze acquisite

7.4 Valutazione personale

Appendice A

Appendice A

Citazione

---

Autore della citazione



# Acronimi e abbreviazioni

**API** [Application Program Interface](#). [1](#), [13](#)

**UML** [Unified Modeling Language](#). [5](#), [13](#)

# Glossario

**API** in informatica con il termine *Application Programming Interface API* (ing. interfaccia di programmazione di un'applicazione) si indica ogni insieme di procedure disponibili al programmatore, di solito raggruppate a formare un set di strumenti specifici per l'espletamento di un determinato compito all'interno di un certo programma. La finalità è ottenere un'astrazione, di solito tra l'hardware e il programmatore o tra software a basso e quello ad alto livello semplificando così il lavoro di programmazione. [12](#)

**UML** in ingegneria del software *UML, Unified Modeling Language* (ing. linguaggio di modellazione unificato) è un linguaggio di modellazione e specifica basato sul paradigma object-oriented. L'*UML* svolge un'importantissima funzione di “lingua franca” nella comunità della progettazione e programmazione a oggetti. Gran parte della letteratura di settore usa tale linguaggio per descrivere soluzioni analitiche e progettuali in modo sintetico e comprensibile a un vasto pubblico. [12](#)

# Bibliografia

## Riferimenti bibliografici

James P. Womack, Daniel T. Jones. *Lean Thinking, Second Editon*. Simon & Schuster, Inc., 2010 (cit. a p. [1](#)).

## Siti web consultati

*Manifesto Agile*. URL: <http://agilemanifesto.org/iso/it/> (cit. a p. [1](#)).