

NOME DELLA TUA UNIVERSITÀ

NOME DEL CORSO DI LAUREA

Ingegneria del Software

Federico Brutti
federico.brutti@studenti.univr.it

Inserire citazione inerente alla materia

Indice

1 Introduzione	5
1.1 Tipi di software	5
1.2 Caratteristiche dei software	5
1.3 Aspetti etici	5
2 Processi Software e sviluppo agile	7
2.1 Modelli dei processi software	7
2.2 Attività dei processi	7
2.3 Gestione dei cambiamenti	7
2.4 Metodi agili	7
2.5 Metodologia SCRUM e relative applicazioni	7
3 Ingegneria dei requisiti	9
3.1 Requisiti funzionali e non funzionali	9
3.2 Processi di ingegneria dei requisiti	9
3.3 Caso di utilizzo UML	9
4 Progettazione orientata agli oggetti	11
4.1 Modellazione di sistemi OO con UML	11
4.2 Fasi della progettazione	11
5 Pattern di progettazione	13
5.1 Pattern architetturali	13
5.2 Pattern di design	13
6 Verifica e Validazione	15
6.1 Ispezione e test	15
6.2 Unit testing	15
6.3 Integration testing	15
6.4 User acceptance testing	15
6.5 Sviluppo guidato dal testing	15

<i>INDICE</i>	3
---------------	---

7 Evoluzione e mantenimento software	17
7.1 Gestione del cambiamento	17
7.2 Sistemi legacy	17
7.3 Manutenzione del software	17
8 Strumenti e framework utili	19
8.1 Visualizzazione di dati - Plotly	19
8.2 Manipolazione e analisi di dati - Pandas	19
8.3 Costruzione di applicazioni analitiche	19
8.3.1 Dash	19
8.3.2 Pony ORM	19
8.4 Applicazioni web complesse	19

Capitolo 1

Introduzione

definizione di ingegneria del software

1.1 Tipi di software

1.2 Caratteristiche dei software

1.3 Aspetti etici

Capitolo 2

Processi Software e sviluppo agile

2.1 Modelli dei processi software

2.2 Attività dei processi

2.3 Gestione dei cambiamenti

2.4 Metodi agili

2.5 Metodologia SCRUM e relative applicazioni

Capitolo 3

Ingegneria dei requisiti

- 3.1 Requisiti funzionali e non funzionali**
- 3.2 Processi di ingegneria dei requisiti**
- 3.3 Caso di utilizzo UML**

Capitolo 4

Progettazione orientata agli oggetti

4.1 Modellazione di sistemi OO con UML

class diagram, use case diagram, activity diagram, sequence diagram

4.2 Fasi della progettazione

Capitolo 5

Pattern di progettazione

5.1 Pattern architetturali

MVC, repository, client server, pipe and filter

5.2 Pattern di design

singleton, observer, template, iterator, factory, abstract factory, proxy, façade, decorator

Capitolo 6

Verifica e Validazione

- 6.1 Ispezione e test**
- 6.2 Unit testing**
- 6.3 Integration testing**
- 6.4 User acceptance testing**
- 6.5 Sviluppo guidato dal testing**

Capitolo 7

Evoluzione e mantenimento software

7.1 Gestione del cambiamento

7.2 Sistemi legacy

7.3 Manutenzione del software

Capitolo 8

Strumenti e framework utili

8.1 Visualizzazione di dati - Plotly

basi di Plotly per creare visualizzazioni di dati interattive.

8.2 Manipolazione e analisi di dati - Pandas

DataFrame di Pandas per una manipolazione e analisi efficiente dei dati.

8.3 Costruzione di applicazioni analitiche

8.3.1 Dash

framework Python per la costruzione di applicazioni web analitiche.

8.3.2 Pony ORM

Pony ORM per semplificare le operazioni di database in Python.

8.4 Applicazioni web complesse

callback avanzati di Dash, per creare applicazioni web interattive più complesse.