

Strumenti di gestione del ciclo di vita del *software*

Università degli studi di Padova
a.a. 2016/17

Laurea in Informatica

Corso di Ingegneria del Software mod. A

Presenta

Nicola Bertazzo

nicola.bertazzo@gmail.com

Mercoledì 30 Novembre 2016

> Sommario

1. Obbiettivi
2. Processo
3. Esempio
4. Visione in dettaglio
5. Conclusioni

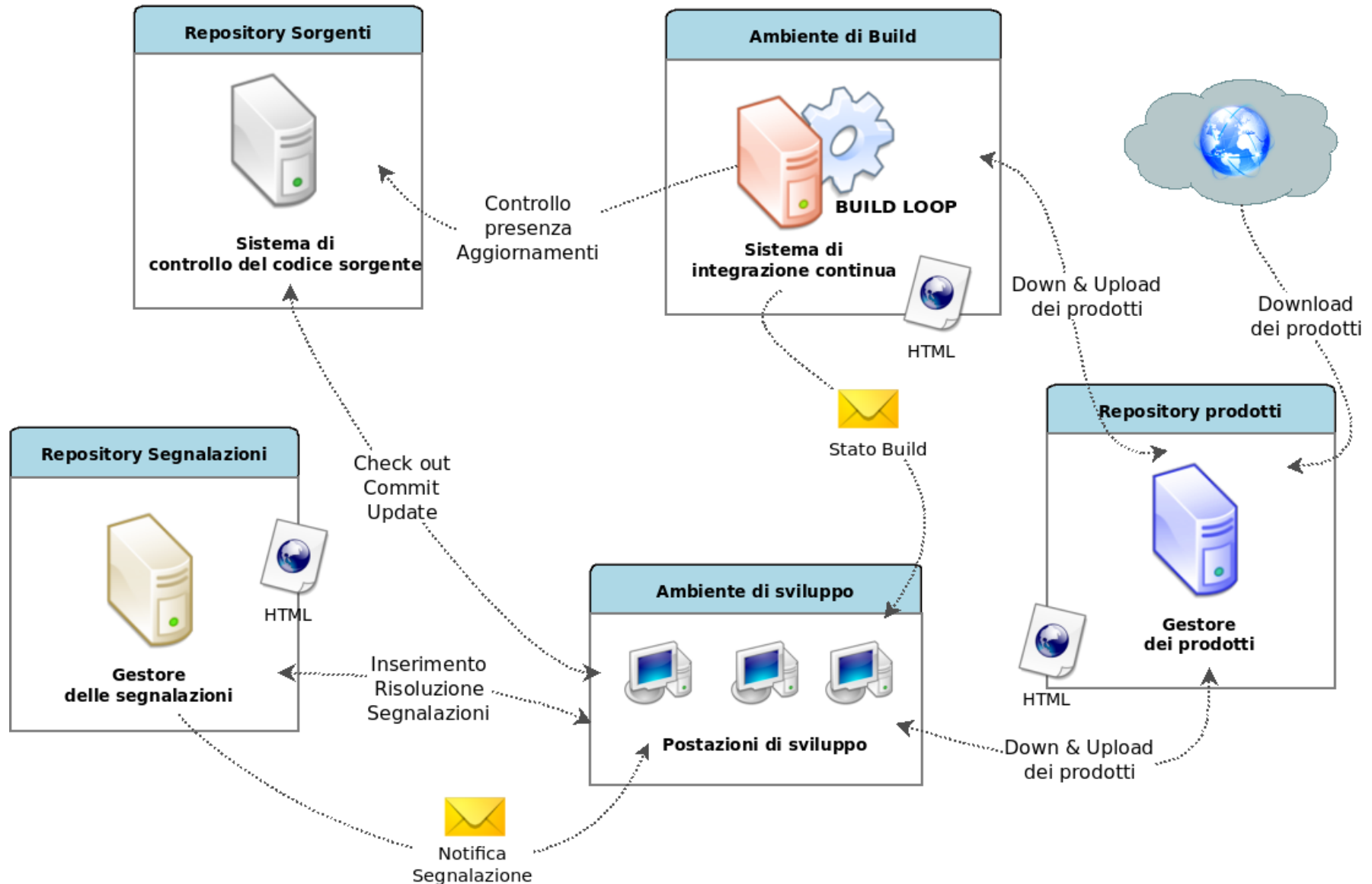
> Obbiettivi

- Descrivere un processo di sviluppo e verifica del software **ripetibile**, con lo scopo di misurare e migliorare la qualità di prodotto
- Fornire un esempio dell'implementazione del processo, per lo sviluppo di un progetto **Java**, con prodotti *Free*

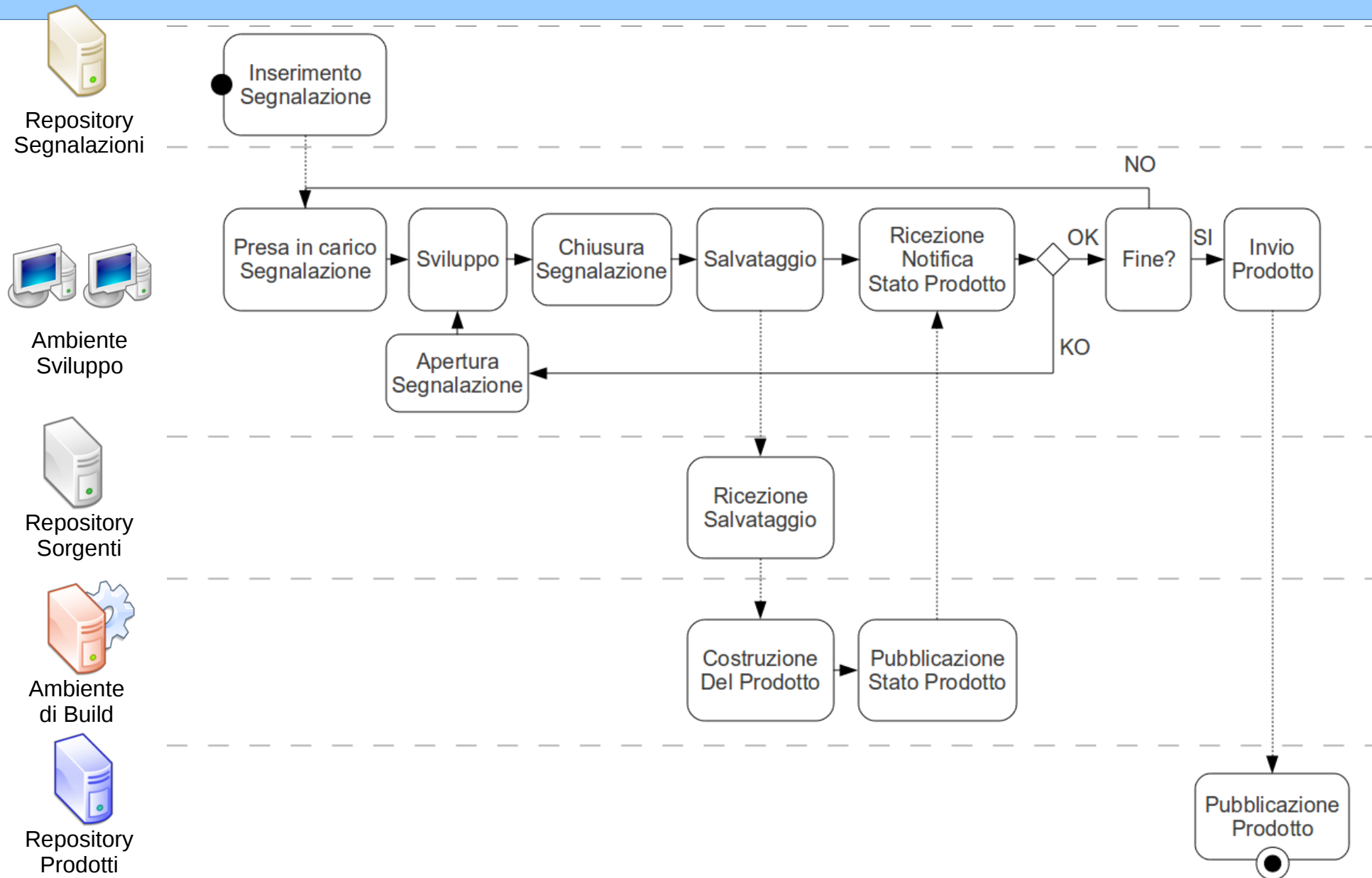


Processo di sviluppo

> Visione generale



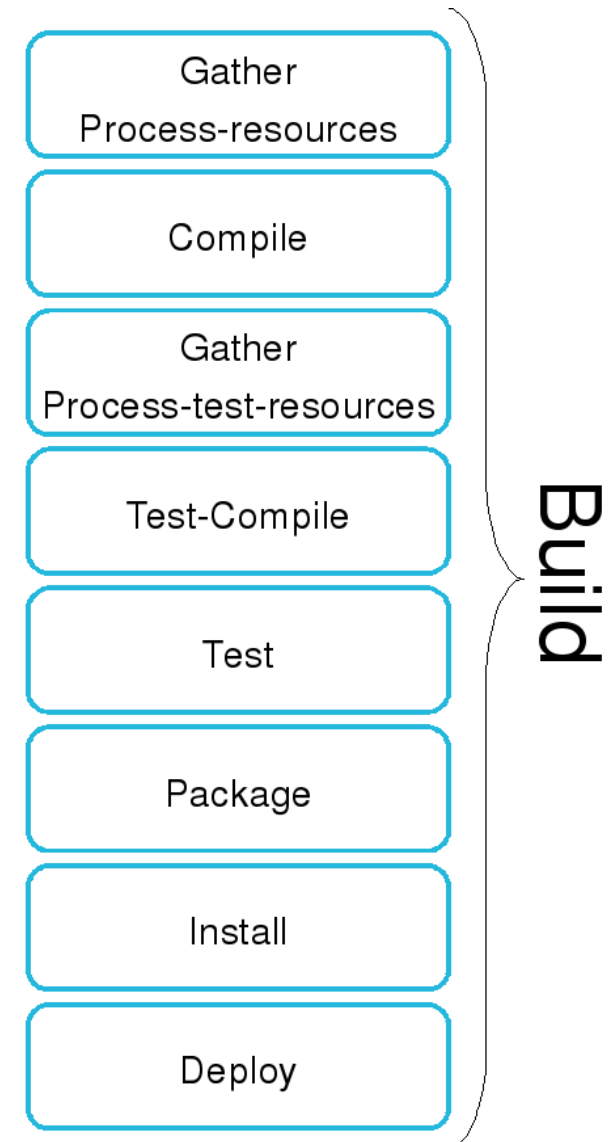
> Esempio



Visione in dettaglio

> Ambiente di sviluppo

- Luogo dove avviene l'attività di sviluppo e verifica del prodotto
- Vengono realizzati i test di unità
- Il processo di costruzione del prodotto deve essere automatico
- Il codice prodotto viene inviato frequentemente al Repository dei sorgenti
- Dopo la realizzazione di una funzionalità avvengono le altre attività di verifica



Ambiente di sviluppo

> Strumenti di verifica

Unità: Controllo del comportamento di ogni singolo oggetto in isolamento

- JUnit, TestNG [TDD]

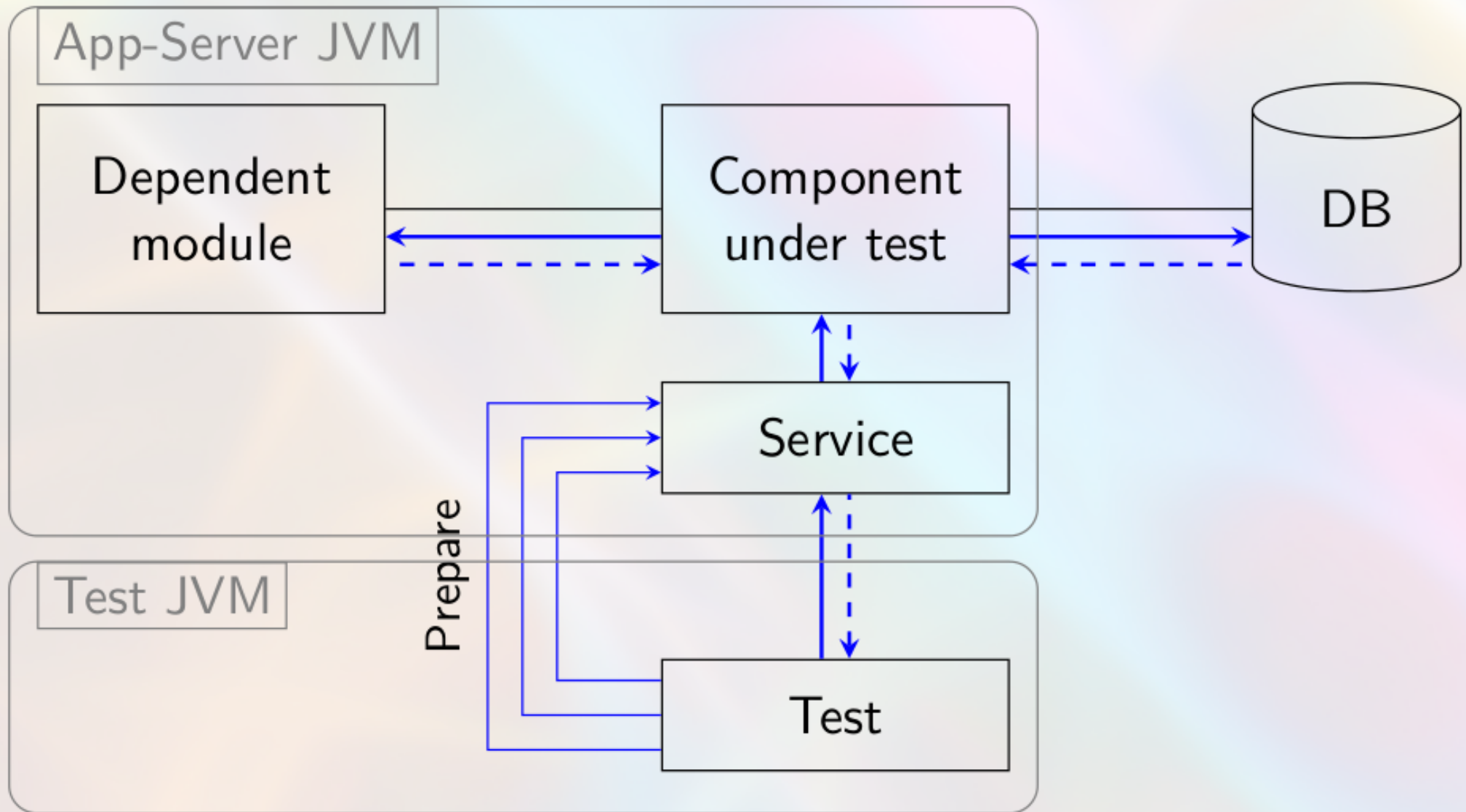
Integrazione: Verifica della collaborazione fra più oggetti nel formare un sottosistema, al fine di verificare il comportamento esterno e i contratti di interfaccia

Funzionali e verifica e validazione: Controllo del soddisfacimento dei requisiti funzionali

- Fitnesse, Selenium, Jubula, Cucumber [BDD]

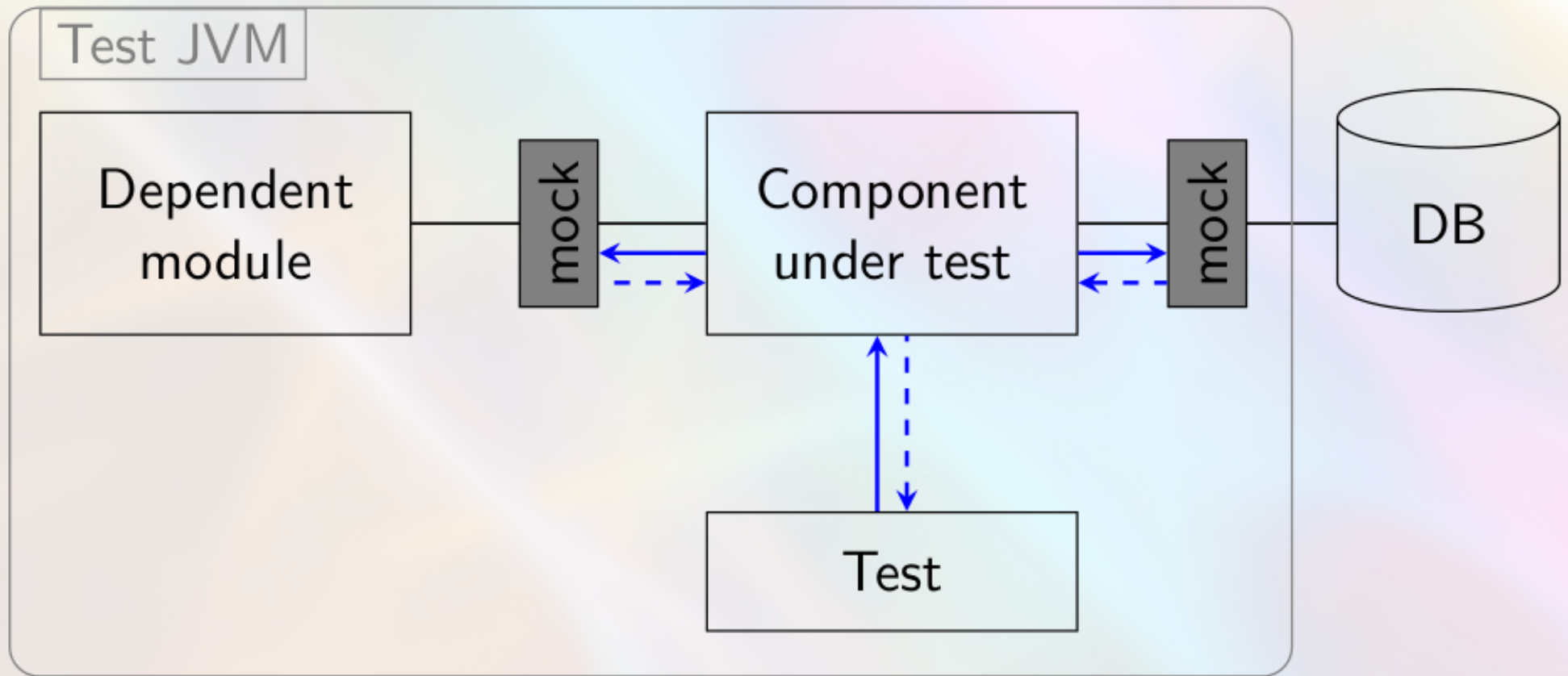
Ambiente di sviluppo

> Test funzionali



Ambiente di sviluppo

> Test di unità



Ambiente di sviluppo

> Strumenti di analisi del codice

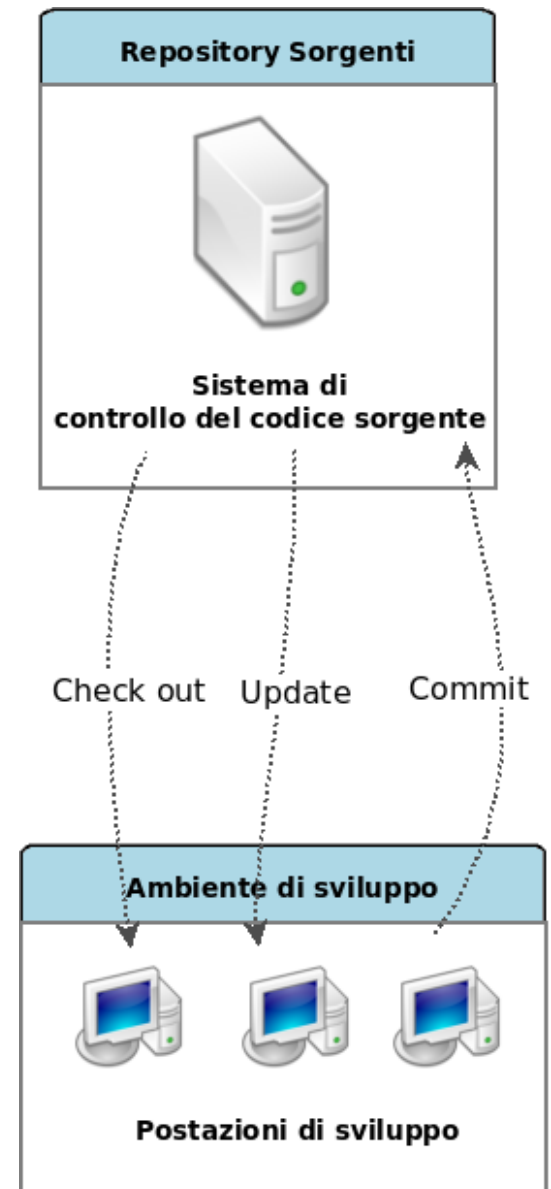
- Imporre il rispetto di convenzioni e stili
- Verificare la congruità della documentazione
- Controllare metriche ed indicatori (complessità ciclomatica, grafo delle dipendenze, numerosità delle linee di codice)
- Ricercare codice copiato in più punti
- Ricercare errori comuni nel codice
- Misurare la percentuale di codice testato
- Ricercare indicatori di parti incomplete (p. es. tag)



Visione in dettaglio

> Repository dei sorgenti

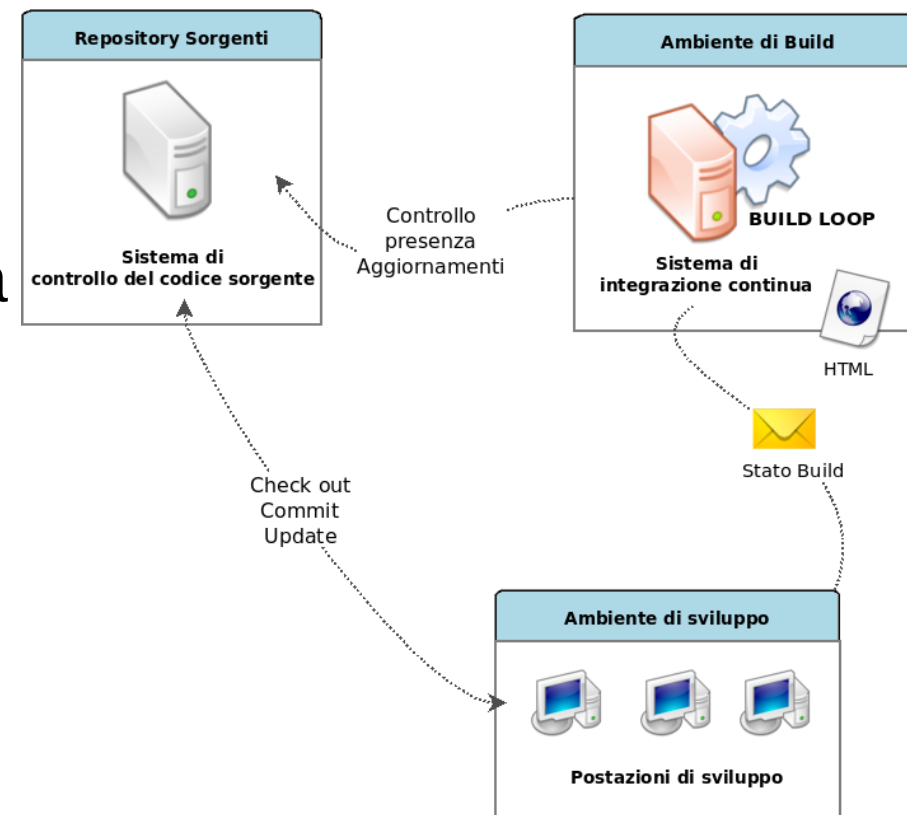
- Deposito unico di tutto il codice sorgente
- Accesso alla storia completa del software
- Documentazione delle modifiche:
 - Oggetti modificati
 - Data e ora
 - Autore
 - Commenti e motivazioni
- Possibilità di gestire più rami diversi di sviluppo contemporaneamente



Visione in dettaglio

> Ambiente di build

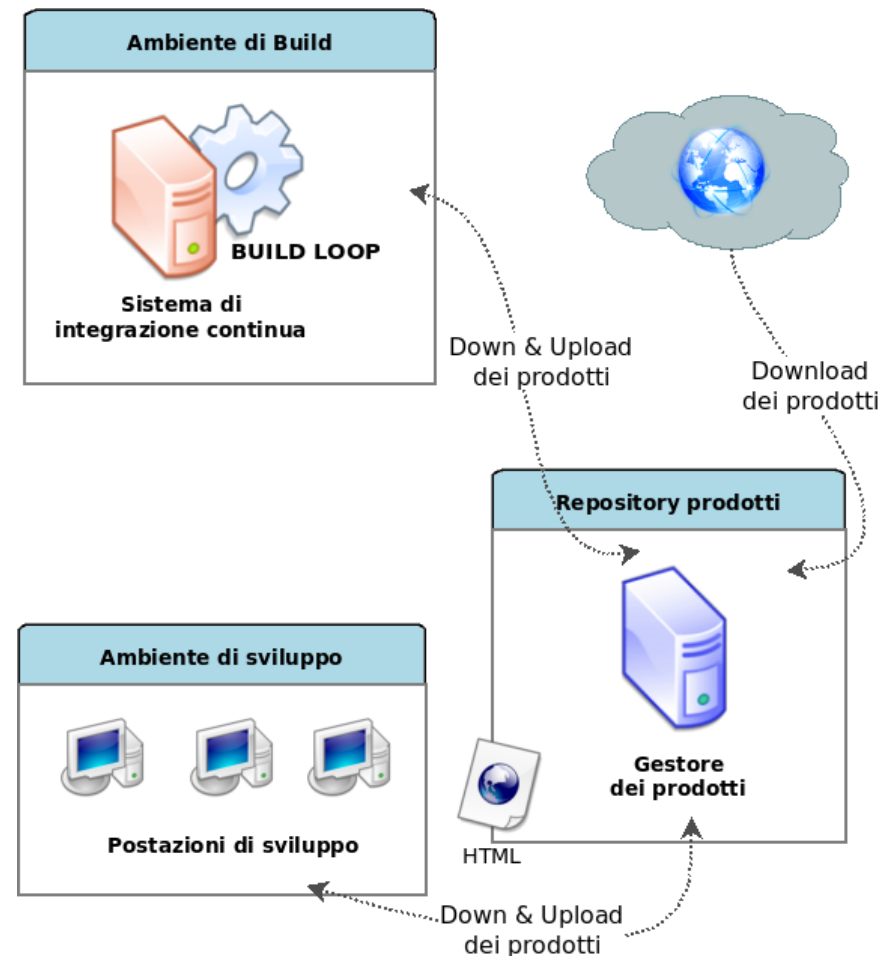
- Al completamento di un'attività viene costruito il prodotto
- Se il processo di costruzione fallisce l'attività non continua fino a che il prodotto non viene riparato
- Assicura la presenza di un prodotto consistente
- Il processo di costruzione deve essere automatico e ripetibile



Visione in dettaglio

> Repository prodotti

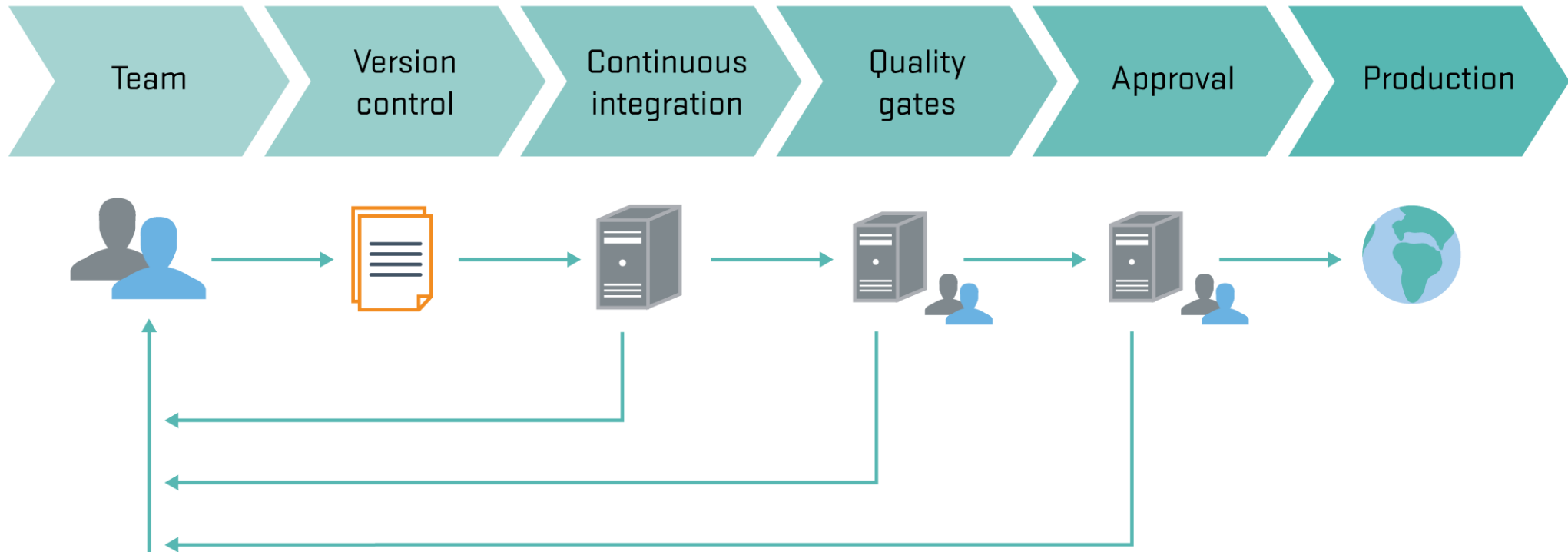
- Ambiente dove depositare e pubblicare i prodotti
- Agisce da intermediario per scaricare prodotti da depositi esterni
- Permette di effettuare ricerche e reperire informazioni riguardanti i prodotti
- Permette di gestire e associare permessi d'accesso sui prodotti



> Conclusioni

- Risultati garantiti nel lungo periodo
- Benefici incrementali
 - Si ottengono vantaggi anche da una applicazione parziale o progressiva
- Tempo di avvio lungo
- Gestione complessa di molte componenti

> ... *Continuous Delivery*





BUSINESS



IT



QUEUE TO PRODUCTION



I CAN SHIP MY IDEAS WHENEVER I LIKE!

BUSINESS



CUSTOMER



CONTINUOUS DELIVERY

BY JEZ HUMBLE & DAVID FARLEY

DONE MEANS RELEASED



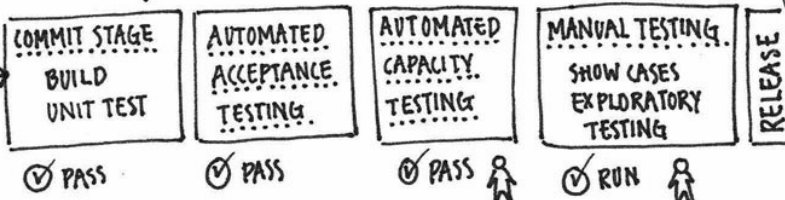
A CLOSER LOOK

COMMIT STAGE

- ✓ CREATING EXECUTABLE CODE MUST WORK. VERIFIES THAT THE SYNTAX OF YOUR SOURCE CODE IS VALID
- ✓ UNIT TEST PASS
- ✓ FULFILL CERTAIN QUALITY CRITERIA SUCH AS TEST COVERAGE AND OTHER TECHNOLOGY-SPECIFIC METRICS

KEY

PATTERN DEPLOYMENT PIPELINE

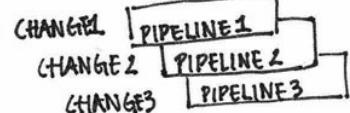


EXAMPLE

CHANGE

CREATE NEW INSTANCE OF PIPELINE

- CHANGE IN
 - EXECUTABLE CODE
 - CONFIGURATION
 - HOST ENVIRONMENT
 - DATA



FEED-BACK

ANY CHANGE IS A TRIGGER • FAST • ACT ON IT

BENEFITS



EMPOWERED - IN CONTROL
LOW STRESS - SMALL RELEASES



REDUCING ERRORS
- CONFIG MGT.
- VERSION CONTROL



DEPLOYMENT FLEXIBILITY
- EASY TO START APPLICATION IN NEW ENVIRONMENT



PRACTICE MAKES PERFECT

SEEMS LIKE THE AUTHORS CAN'T STRESS IT ENOUGH. IT'S EVERYWHERE THROUGHOUT THIS BOOK..



AUTOMATE ALMOST EVERYTHING

VERSION CONTROL



“

ENCOURAGING GREATER COLLABORATION BETWEEN EVERYONE INVOLVED IN SOFTWARE DELIVERY IN ORDER TO RELEASE VALUABLE SOFTWARE FASTER AND MORE RELIABLY.

”

If it hurts, do it more frequently

CC BY-SA

Nhan Ngo