



UNICAM
UNIVERSITÀ DI CAMERINO



IV. UP and Use Case Definitions

Objectives

- ▼ To provide a general overview of the Unified Process
- ▼ To describe Use Cases
- ▼ Exercises on Use Case Definition

UP Phases

- ▼ The UP is an **iterative process** consisting of 4 different phases:
 - **Inception**
 - **Elaboration**
 - **Construction**
 - **Transition**
- ▼ In each of this phase one or more of the development steps described in last week lessons are carried on
- ▼ The Rational Unified Process is a kind of UP supported by tools developed and distributed by Rational.

Why iterative development?

- ▼ UP defined from experienced software engineer
 - **Requirements change during development!!**
 - **Clients do not know what they exactly want!!**
- ▼ Restating Iterative process benefits:
 - Early mitigation of high risk
 - Early visible progress
 - Early feedback, user engagement, and adaptation, leading to a refined system that more closely meets the real needs of the stakeholders
 - Managed complexity; the team is not overwhelmed by “analysis paralysis” or very long and complex steps
 - The learning within an iteration can be methodically used to improve the development process itself, iteration by iteration

(from “Applying UML and Patterns”, C. Larman – Prentice Hall)

Inception

- ▼ This is a short phase (not more than few weeks) and it is useful to:
 - Approximate vision
 - Business case
 - Scope
 - Vague Estimates

Elaboration

- ▼ Main scope of this phase are:
 - To refine the vision
 - Starts the iterative implementation of the core architecture
 - To find solution to address high risks
 - Identification of most of the requirements and scope
 - To provide realistic estimates

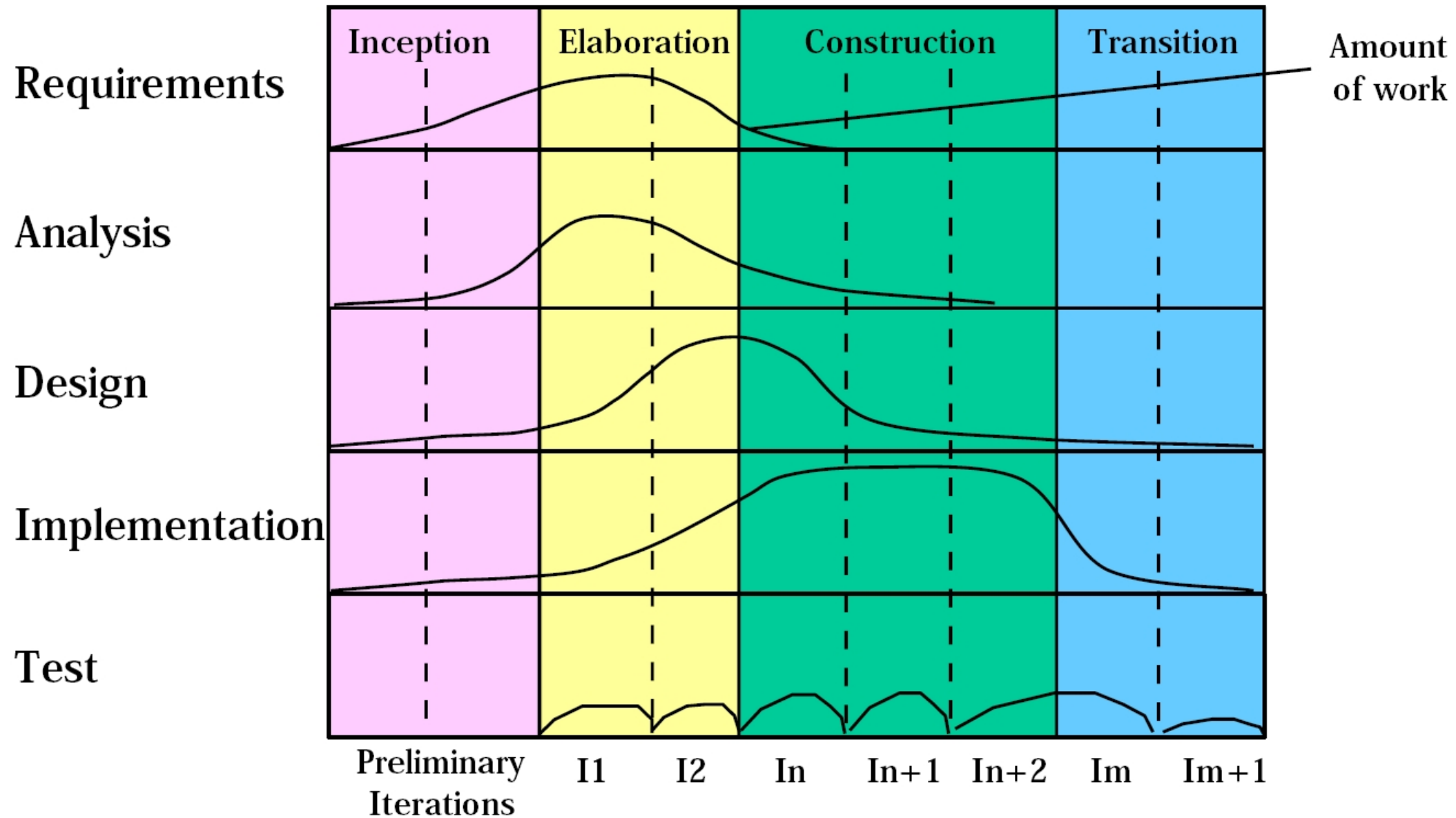
Construction

- ▼ Main objective of this phase are:
 - Iterative development of the remaining and less risky functionality
 - Prepare the system for final deployment

Transition

- ▼ Main objectives of this phase are:
 - Beta test the system
 - Deploy the system in the final environment

UP and “Standard” Development Phases



Iteration Length and Timeboxing

▼ Length of an iteration:

- For each iteration UP recommends a **duration of 2-6 weeks**
- Objective is to have small steps and rapid feedbacks

▼ Iteration are **timeboxed**

- The iteration **must** end with something working
- If it is difficult to have a stable system by the deadline is better to **discard functionality**
- Discarded functionality can be scheduled within the next iteration

UP principles

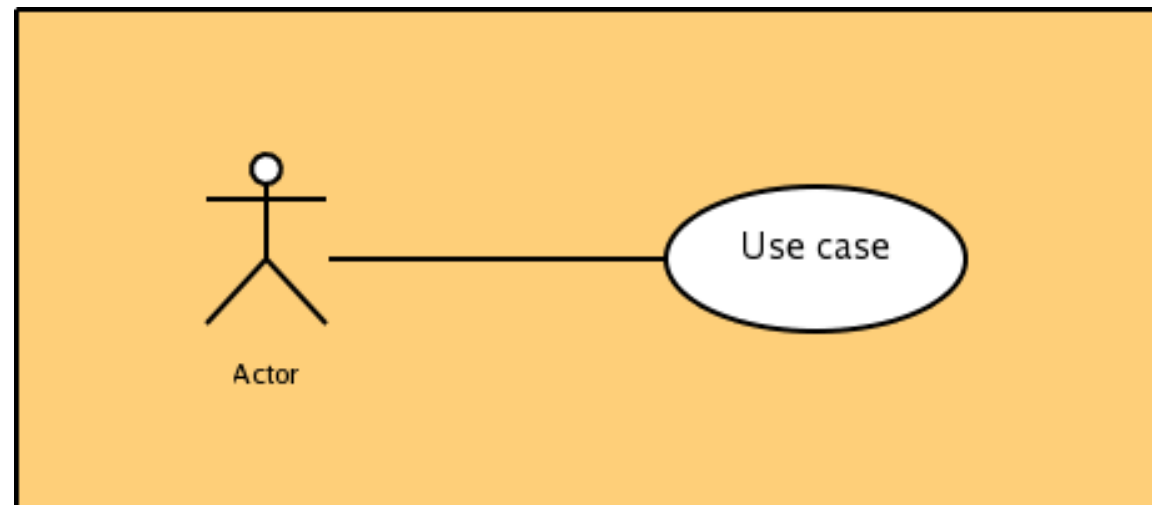
- ▼ **High risk** and **high value** issues must be addressed in **early iterations**
- ▼ **Users** must be **continuously engaged**
- ▼ Build a cohesive, **core architecture in the early iterations**
- ▼ Quality must be continuously verified
- ▼ **Apply use cases**
- ▼ Use **graphical notations**
- ▼ Requirements must be carefully managed

Use-Case Models

- ▼ Widely and used mechanism to **discover and record requirements**
- ▼ Approach suggested by UP and supported by Unified Modeling Language (UML)
- ▼ UML ...

What Use cases are?

- ▼ Use cases are a mechanisms to help keep it simple and understandable for all stakeholders. Informally they are **stories of using a system to meet goals**
- ▼ One of the main strength of this mechanism is the **capacity to scale up and down in terms of sophistication and formality**, depending on need
- ▼ UML notation:



Black Box use cases

- ▼ BB use cases do **not** describe **the internal of the system**, its components, or design. Rather the system is described as having **responsibilities**

Black Box Style

The system records all product sale

Not

The system write the sale in a list data structure

Or even worse....

The system generate a list.insert(sale) for the sale

Use Cases...

- ▼ Use cases are carried on by **actors**:
something with behaviour, such as a person, computer system, or organisation
 - **Primary actor** – has goals fulfilled through using services of the SuD.
 - **Supporting actor** – provide a service to the SuD. Often a computer system.
 - **Offstage actor** – has interest in the behavior of the system but is not primary or supporting (e.g. Tax agency)
- ▼ A **scenario** is a specific sequence of actions and interactions between actors and the system under development
- ▼ A use case is then a collection of **related success and failure scenarios**

Organising Use cases

- ▼ Use cases should be grouped with relation to functionality
- ▼ Use cases can be reused:
 - Inclusion (e.g. Entering PIN for withdrawals and check balance)
 - Extension (extension points)
 - Generalization/Specialization (refining the flow of events)



A complex use case - POS

Vendita

Primary Actor: Cassiere

Stakeholders and Interests:

- cassiere: desidera un sistema affidabile e veloce. Errori gli possono essere addebitati
- addetto alle vendite: vuole che le commissioni siano calcolate correttamente
- cliente: vuole acquistare prodotti velocemente. Vuole un dettaglio degli acquisti per verifica.
- Società: vuole che tutte le transazioni siano memorizzate e che il cliente sia soddisfatto. Vuole certezza nelle transazioni con pagamenti bancomat
- governo: vuole che le imposte/tasse (e.g. IVA) siano calcolate correttamente sui prodotti
- Circuito bancomat: vuole ricevere dati in formato corretto, e vuole che tutte le transazioni siano corrette e non contestabili

Precondizioni: Il cassiere è identificato e autenticato

Postcondizioni: La vendita è registrata nel sistema. Oneri fiscali calcolati correttamente. Inventario è aggiornato. Commissioni per l'agente sono registrate. La ricevuta per il cliente è prodotta. L'autorizzazione al pagamento è stata ricevuta e registrata.

Venditacontinua

Principale scenario di successo:

1. Il cliente arriva alla cassa con i prodotti e servizi da acquistare.
2. Il cassiere inizia una nuova transazione di vendita
3. Il cassiere inserisce l'identificatore del prodotto
4. Il sistema registra la vendita del singolo prodotto e aggiunge il prodotto alla ricevuta
il cassiere ripete 3 e 4 finché non decide di fermarsi
5. Il sistema presenta il totale inclusivo delle tasse calcolate
6. il cassiere comunica il totale al cliente e richiede il pagamento
7. Il cliente paga ed il cassiere gestisce il pagamento
8. Il sistema registra la transazione ed invia informazioni al sistema di pagamento e al sistema di inventario
9. Il sistema produce la ricevuta
10. Il cliente lascia l'esercizio con i prodotti e la ricevuta

A complex use case - POS

Venditacontinua

Estensioni:

*a. In qualsiasi momento il sistema potrebbe andare in errore:

Assicurarsi che il sistema supporti l'evenienza e preveda meccanismi di fault-tolerance e recovery delle transazioni. Il sistema non deve confermare transazioni incomplete.

1. Il cassiere fa ripartire il sistema e richiede il recovery

2. il sistema ricostruisce lo stato precedente

2a. il sistema rileva anomalie che non permettono il recupero della transazione

1. il sistema segnala l'errore al cassiere, lo registra, e riparte da uno stato corretto

2. il cassiere riparte con le transazioni che non sono andate a buon fine

5b. Il cliente comunica che ha diritto a particolari sconti

1. il cassiere segnala al sistema la richiesta di sconto

2. il cassiere inserisce l'identificatore del cliente

3. il sistema produce il nuovo totale

6a. Il cliente dice che vuole pagare in contanti ma si accorge di non avere abbastanza contante

1a. Il cliente chiede un metodo alternativo di pagamento

1b. Il cliente chiede di annullare la transazione.

Il cassiere annulla la transazione ed il sistema non procede alla registrazione.

A complex use case - POS

Venditacontinua

Requisiti speciali:

- Il sistema deve essere fornito di un touch screen . Il testo deve essere leggibile ad una persona senza problemi di vista da almeno un metro
- la transazione deve essere conclusa entro 30 secondi nel 90% dei casi

....

Technologie da poter inserire nel sistema:

3a. Prodotto riconosciuto tramite lettura laser del codice a barre

7a. Firma pagamento carta di credito su carta ma il sistema deve poter supportare estensione a firma supporto elettronico

Frequenza del caso d'uso: praticamente in continuo

Problemi da investigare:

- regolamento per la tassazione e supporto a variazione
- recupero informazioni a seguito di fallimento di comunicazioni con sistemi remoti
- ...

Element in a Use case definition

- ▼ Preface
- ▼ Attori e lista degli obiettivi
- ▼ Precondizioni e postcondizioni
- ▼ Principale scenario di successo
- ▼ Estensioni – (si compone di condizione e gestione)
- ▼ Requisiti speciali
- ▼ Technologie



Elementary Business processes

- ▼ Guideline: define Use Cases at the **level of EBP**
- ▼ EBP: A task performed by one person in one place at one time, in response to a business event, which add measurable business value and leaves the data in a consistent state.
- ▼ The main scenario should be **around 5-10 lines long**

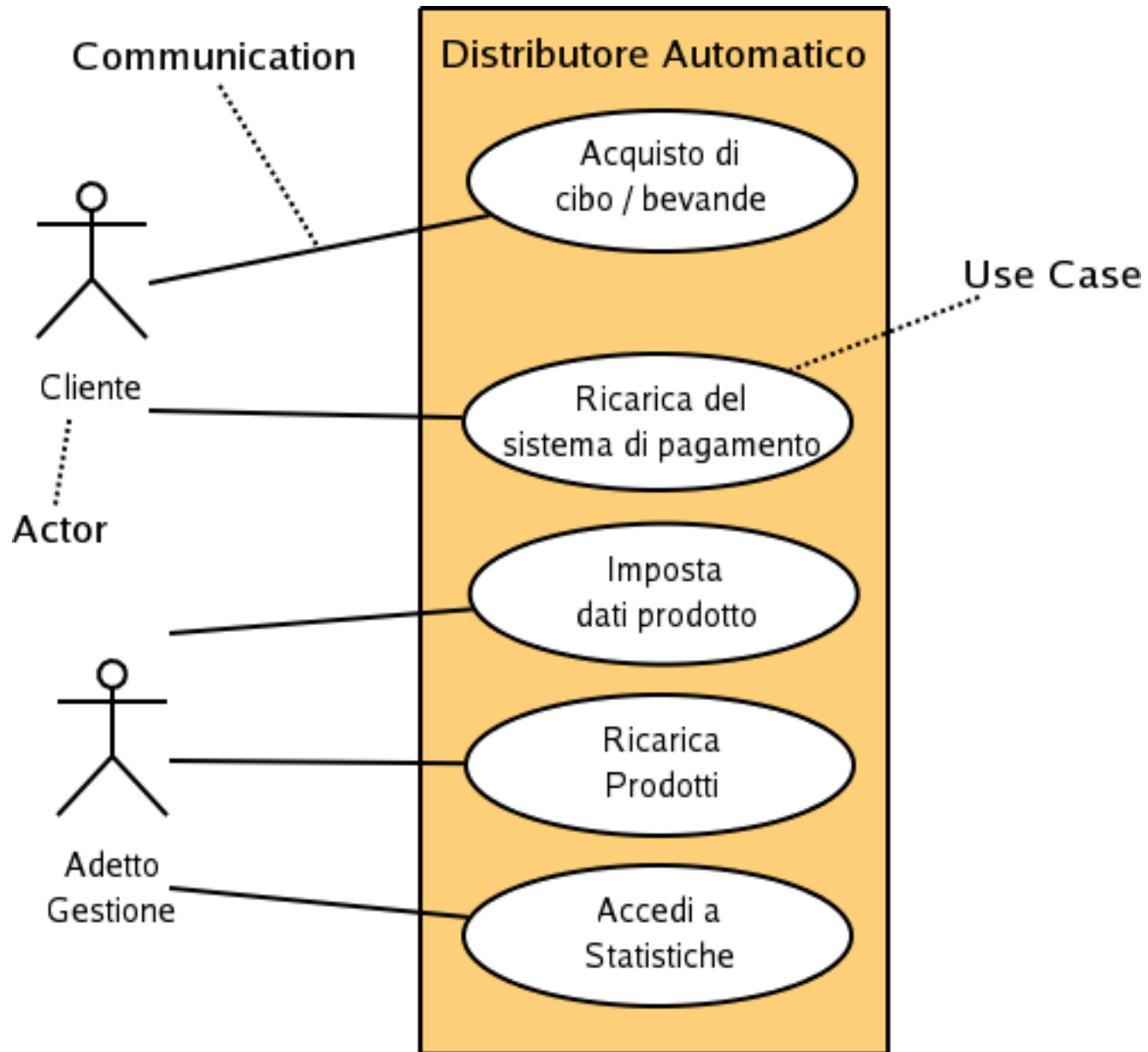


A Use Case Elicitation Process

- ▼ Choose the system boundary
- ▼ Identify the **primary actors**
- ▼ For each actor identify goals
- ▼ Define use cases satisfying defined goals



Representing System Use Cases



Use cases and UP

- ▼ After the Inception phase around 10%-20% of core functionality use cases are generally defined
- ▼ After the Elaboration phase around 80%-90% are written in detail
- ▼ During Construction the remaining minor use cases are identified and defined. During this phase the majority of requirements are mature enough and quite stable



Exercises

▼ The on-line box office

▼ Actors:

- Customer
- Event organiser
- Box office manager
- ...

