ANALYSIS OF IT PROJECTS

PROJECT MANAGEMENT METHODOLOGIES PART I

ELSA ESTEVEZ
Universidad Nacional del Sur
Universidad Nacional de La Plata
CONICET, Argentina

AIM AND AGENDA

GDAŃSK UNIVERSITY OF TECHNOLOGY

AIM

To present and compare project management methodologies.

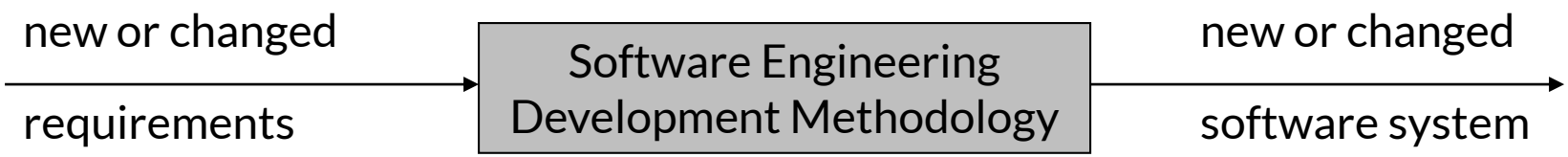
AGENDA

1	PROCESS	What is software development process and what are agile methods?
2	SCRUM	What are main features of SCRUM?
3	RUP	What are main features of RUP?
4	SUMMARY	What was covered in this section?

WHY A METHODOLOGY ?

A methodology for software development contributes to define:

1) who is doing what
2) when to do it
3) how to reach a certain goal
4) the inputs and outputs for each activity



SOFTWARE DEVELOPMENT METHODOLOGY

GDAŃSK UNIVERSITY OF TECHNOLOGY

It is a framework which guides the tasks, people and defines output products of a software development process.

It is a framework because:

1) provides the inputs and outputs of each activity
2) does not restrict how each activity must be performed
3) must be tailored for every project

There is no universal methodology for all kinds of projects.

AGILE DEVELOPMENT

“Agile Software Development is a paradigm of software development methodologies based on agile processes”

"Agile development" is a term derived from the Agile Manifesto, written in 2001 by a group that included the creators of Scrum, Extreme Programming (XP), Dynamic Systems Development Method (DSDM) and Crystal.

Jeff Sutherland

AGILE METHODOLOGIES

The development cycle applied by Agile Methodologies is iterative and incremental, allowing the software to be delivered in small and usable parts, known as increments.

Each iteration can be considered as a small project where activities such as requirement, analysis, design, implementation and testing are carried out with the objective of producing a subset of the final system.

The process is repeated several times producing a new increment in ever cycle until the complete product is finished.

Although all the agile methodologies adopt this cycle, each one of them presents its own characteristics.

AGILE METHODOLOGIES - VALUES

1) Value individuals and their interaction more than processes and tools.
2) Value the software that works over an exhaustive documentation.
3) Value collaboration with the client more than contract negotiation.
4) Value more the response to change than following a plan.

1 – VALUEING INDIVIDUALS

GDAŃSK UNIVERSITY OF TECHNOLOGY

Value individuals and their interaction more than processes and tools.

Three premises sustain the principle:

- Team members are the main factor of a project's success
- It's more important to set up a team than the environment.
- It's better to put a team together and to let it configure the environment based on its own needs.

2 – VALUEING SOFTWARE

GDAŃSK UNIVERSITY OF TECHNOLOGY

Value the software that works over an exhaustive documentation.

The principle is based on the premise that documents can neither replace nor offer the added value that is achieved with direct communication between people through the interaction with prototypes.

The use of documentation that generates works and does not add a direct value to the product must be reduced to the minimum essential.

3 – VALUEING COLLABORATION

GDAŃSK UNIVERSITY OF TECHNOLOGY

Value collaboration with the client more than contract negotiation.

In agile development, the customer is integrated and collaborates with the development team, just like any other member.

The contract itself does not add value to the product; it is just a formalism that establishes lines of responsibility among parties.

4 –VALUEING RESPONSE TO CHANGE

GDAŃSK UNIVERSITY OF TECHNOLOGY

Value more the response to change than following a plan.

The speedy and constant evolution must be inherent factors to the development process.

Value should be given to the ability to react to change over the ability to monitor and assure the fulfilment of pre-established plans.

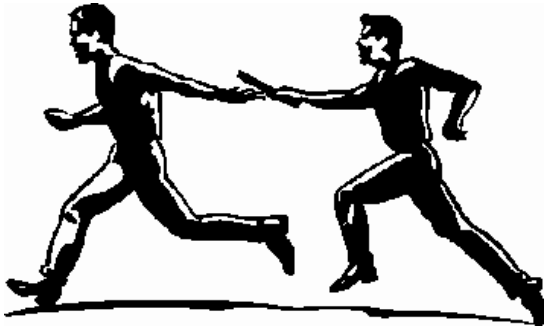
AGILE METHOD - PRINCIPLES

GDAŃSK UNIVERSITY OF TECHNOLOGY

- Customer satisfaction through early and continuous software delivery
- Accommodate changing requirements throughout the development process
- Frequent delivery of working software
- Collaboration between the business stakeholders and developers throughout the project
- Support, trust, and motivate the people involved
- Enable face-to-face interactions
- Working software is the primary measure of progress
- Agile processes to support a consistent development pace
- Attention to technical detail and design enhances agility
- Simplicity
- Self-organizing teams encourage great architectures, requirements, and designs
- Regular reflections on how to become more effective

COMPARISON

TRADITIONAL DEVELOPMENT	AGILE DEVELOPMENT
Specialized practitioners	Multi-disciplinary team
Phases	Overlaps
Detailed requirements	Product vision
Detailed planning	Adaptation to changes



AGILE METHODOLOGIES – EXAMPLES

Example of most used agile methdologies include:

- SCRUM*
- eXtreme Programming (XP)*
- Feature Driven Development (FDD)
- Dynamic Systems Development Method (DSDM)
- Adaptive Software Development (ASD)
- Crystal, and Lean Software Development (LSD)

* Most used methodologies

AIM AND AGENDA

AIM

To present and compare project management methodologies.

AGENDA

1	PROCESS	What is software development process and what are agile methods?
2	SCRUM	What are main features of SCRUM?
3	RUP	What are main features of RUP?
4	SUMMARY	What was covered in this section?

SCRUM

It is a framework for agile development processes.

Features:

The development is done through iterations (Sprints).

The result of each of them is an executable increase that is shown to the client.

There are daily 15-mins meetings throughout the project of the development team for coordination and integratio

"as in rugby, the ball gets passed within the team as it moves as a unit up the field"

Takeuchi and Nonaka, "*The new product development game*"

SCRUM – PILLARS

GDAŃSK UNIVERSITY OF TECHNOLOGY

Visibility

The aspects of the process that affect the result should be visible to those who are responsible for the result. For example, whiteboards and other mechanisms or collaborative techniques are used to improve communication.

Inspection

The various aspects of the process must be controlled with sufficient frequency so that unacceptable variations in the process can be detected.

Adaptation

The product must be within acceptable limits. In case of deviation, the process and the processed material will be adapted. This adjustment must be made as soon as possible to minimize major deviations.

SCRUM – PRACTICES

The practices employed by SCRUM to maintain an agile control of projects include:

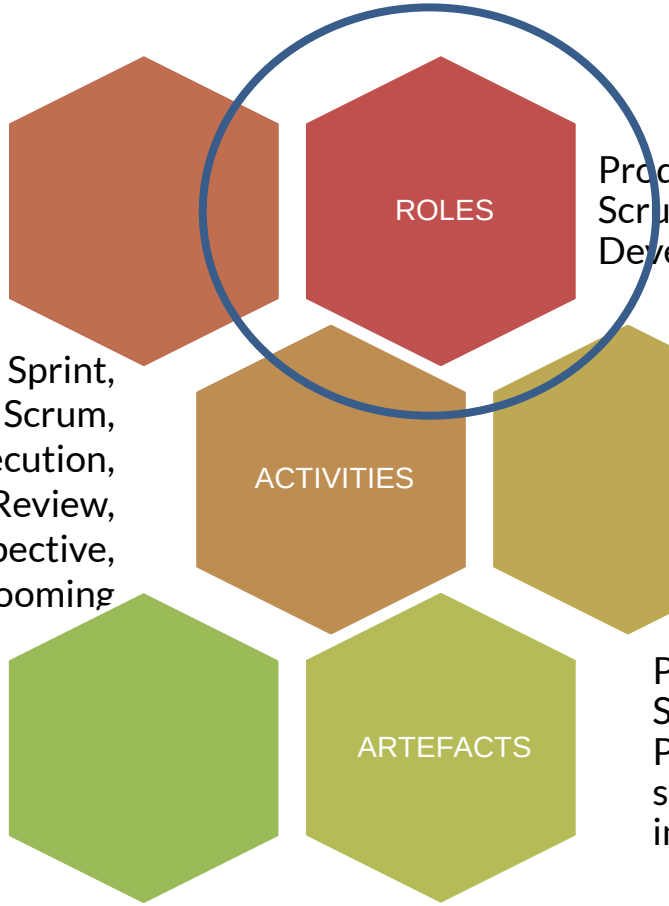
- Incremental development
- Review of iterations
- Self-organized and multi-functional team
- Collaboration
- Prioritization criteria defined by the client

SCRUM – MOTIVATION

Why to use SCRUM?

- is simple
- emphasizes visibility
- uses clear roles and norms
- is based on personal commitment
- solves problems day by day
- allows agile measurements

SCRUM – PRACTICES



Product owner,
Scrum Master,
Development Team

Sprint,
Sprint Planning, Daliy Scrum,
Sprint Execution,
Sprint Review,
Sprint Retrospective,
Product backlog grooming

Product backlog
Sprint backlog
Potentially
shippable product
increment

ROLES – PRODUCT OWNER

Responsible for:

- identifying the functionality
- defining a prioritized list of functionality to be implemented
- deciding which one is first for the next Sprint, reprioritizing and refining them continuously.

He/she represents the client or final user. Usually, the Product Owner and the Client are the same person.

ROLES – SCRUM MASTER

Responsible for:

- ensuring that the team is well represented and is productive
- enabling cooperation between all roles and functions
- eliminating barriers
- isolating and defending the team from external interference
- ensuring that the team and Product Owner follow Scrum.

He/she acts like a facilitator or coach.

ROLES – DEVELOPMENT TEAM

Responsible for:

- selecting the sprint goals
- making the problems visible.

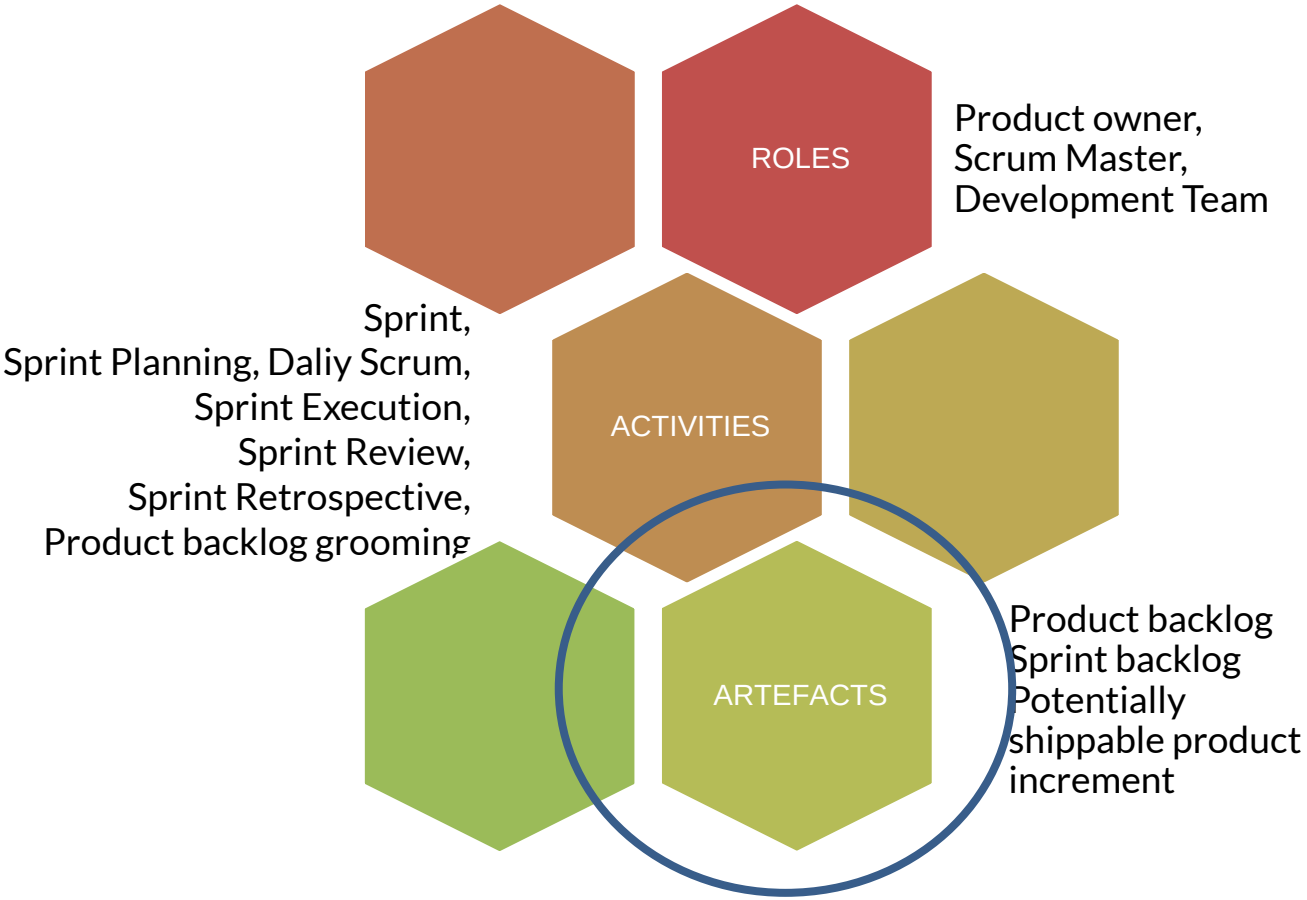
Typically comprising 5 to 10 people from all areas involved in the project, with full time availability.

The team organizes itself.

Members can switch between sprints.

It must include people with the characteristics and skills necessary to meet the goal of the sprint.

SCRUM – PRACTICES



ARTEFACTS

Scrum uses two kind of artefacts for the registration and communication of requirements:

1) Product Backlog – related to business requirements or needs, from the point of view of the client. In the traditional software engineering, is related to the phase of system requirements.

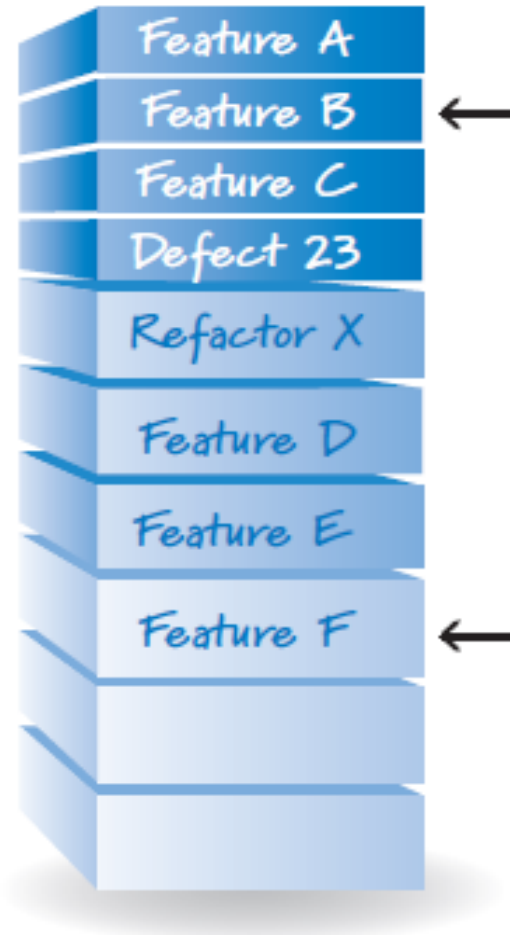
2) Sprint Backlog – related to software specification requirements, necessary to respond to the functionalities expected by the client.

ARTEFACTS – PRODUCT BACKLOG (1)

- It documents system requirements. The starting point is the vision of the product to be obtained, and it evolves during the development.
- It is a registry (inventory) of the features that the product owner wishes to obtain, listed according to the assigned priorities.
- It is a "living" document, in constant evolution.
- It is accessible to all the people involved in the development.
- Everyone can contribute with suggestions.
- The person responsible for the product backlog is the Product Owner.

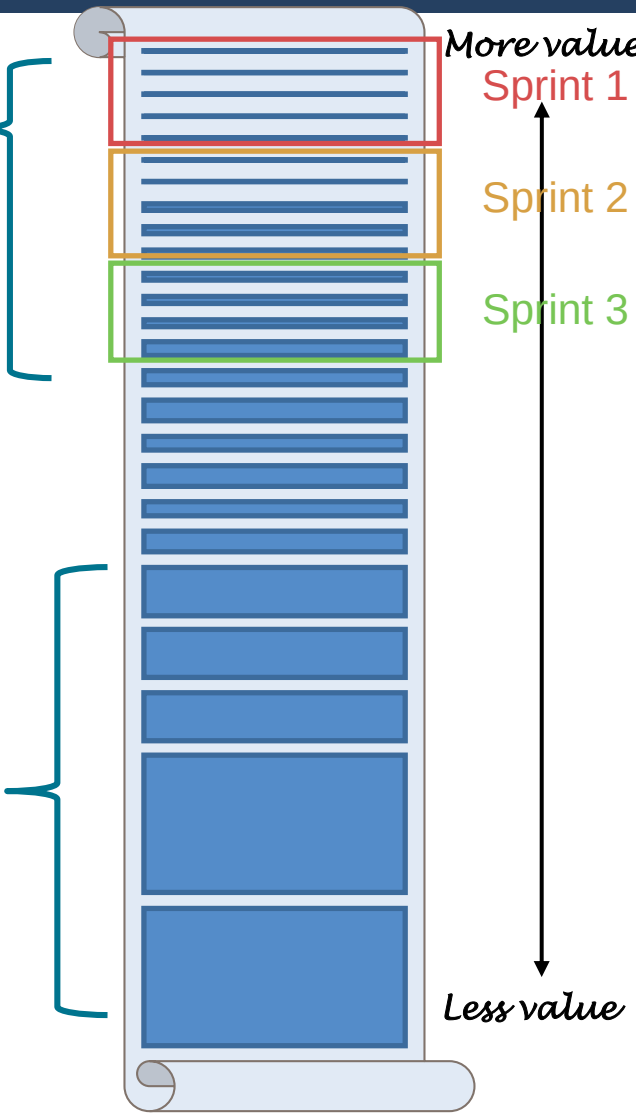
ARTEFACTS – PRODUCT BACKLOG (2)

List of requirements prioritized by their business value and estimated on size



High-priority items

Low-priority items



Each Sprint takes those prioritized

Items less prioritized are less refined

PRODUCT BACKLOG – TEMPLATE

GDAŃSK UNIVERSITY OF TECHNOLOGY

- It is not a requirement document, but a reference tool for the team.

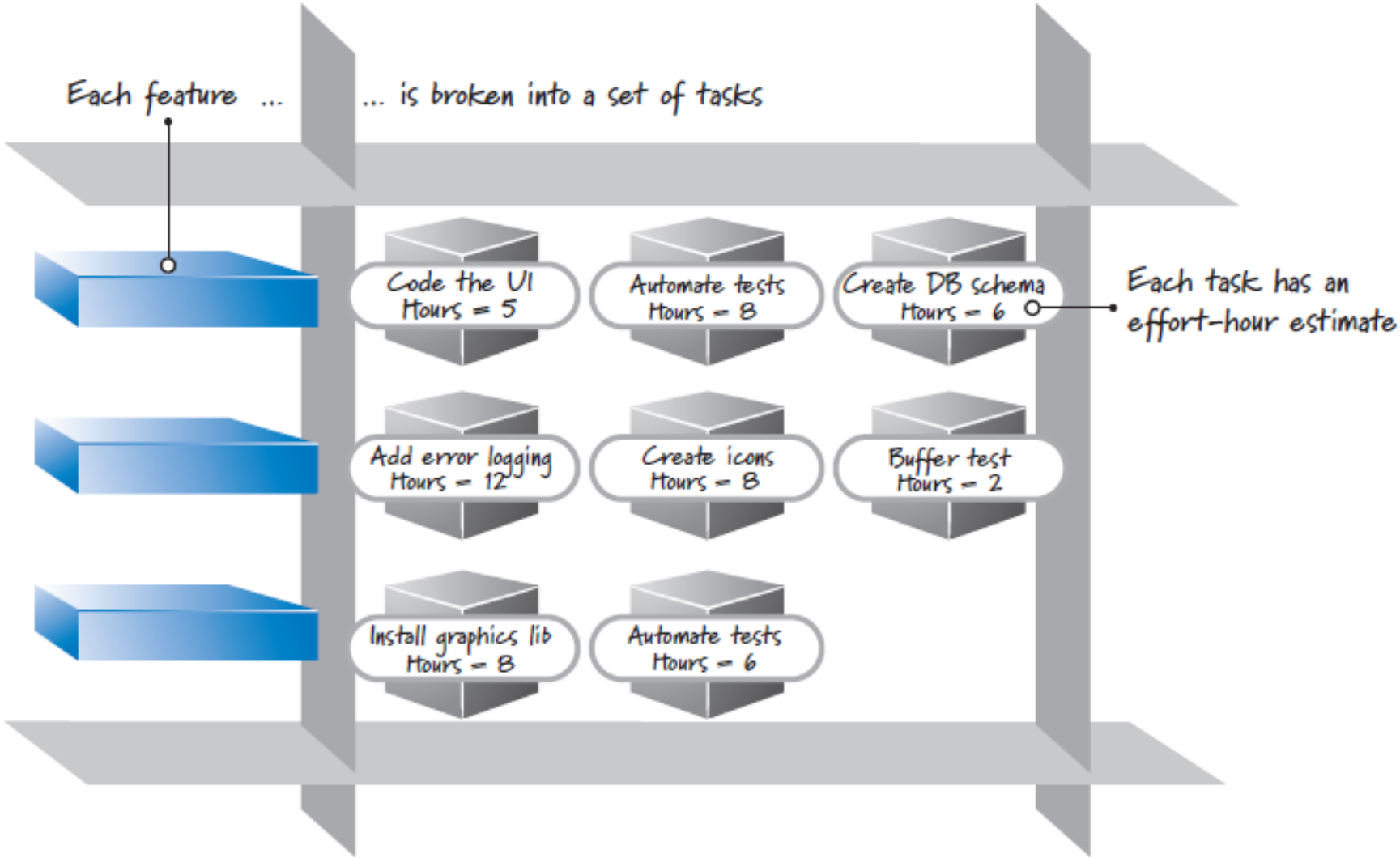
- The document shall include at least the following information for each item:
  - ✓ Unique identifier of the functionality or work.
  - ✓ Description of functionality
  - ✓ Field to indicate the priority
  - ✓ Estimated size

- Depending on the type of project, team organization and culture, other fields may be advisable:
  - ✓ Observations or comments
  - ✓ Validation criterion
  - ✓ Assigned person
  - ✓ Number of Sprint in which it is developed
  - ✓ Module of the system to which it belongs

ARTIFACTS – SPRINT BACKLOG (1)

- It is a list that decomposes the functionalities of the Product Backlog in the tasks needed to build an *increment* - a complete and operating part of the product.
- It also serves to assign to each task the person responsible to carry it out, and indicates the estimated working time to finish it.
- It is useful because it breaks down the project into tasks of adequate size and helps to determine the progress on a daily basis.
- It facilitates the identification of risks and problems without the need for complex management processes.
- It is also a support tool for direct communication among team members.

ARTIFACTS – SPRINT BACKLOG (2)



ARTIFACTS – SPRINT BACKLOG (3)

- The Development Team modifies the Sprint Backlog during the Sprint and the backlog emerges throughout the Sprint.
- As new work is required, the Development Team adds it to the Sprint Backlog.
- As soon as the work is done or completed, the estimated remaining work is updated.
- When some element of the plan is considered unnecessary, it is eliminated.
- Only the Development Team can change their Sprint Backlog during the Sprint.

SPRINT BACKLOG – TEMPLATE

- It includes:
  - ✓ list of tasks,
  - ✓ responsible person of each task,
  - ✓ state in which
  - ✓ remaining time for completion.

- It serves as a support to record in each daily meeting, the time left for each task

- The type of support and model chosen facilitates the consultation and daily and direct communication of the team.

- Possible options for its implementation include:
  - ✓ spreadsheet
  - ✓ blackboard or physical wall
  - ✓ collaborative or project management tool.

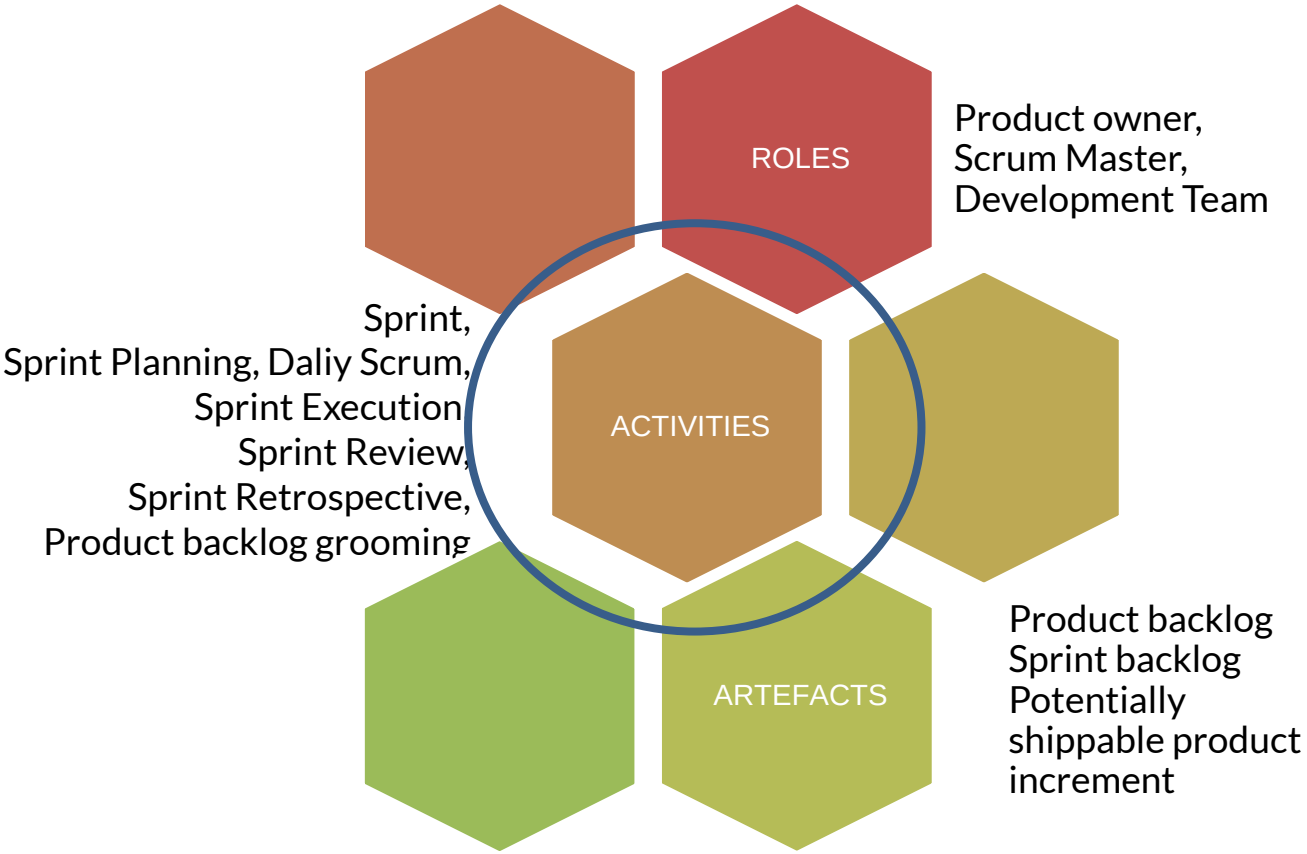
ARTIFACTS – INCREMENT

GDAŃSK UNIVERSITY OF TECHNOLOGY

Potentially shippable product increment

- It is the sum of all the elements of the Product Backlog completed during a Sprint and the value of the increments of all previous Sprints.
- It is a completely finished result, tested and able to be used and delivered to the final customer.
- If the project or system requires documentation, or documented validation and verification processes, or interactions with third-party processes, all of them must be completed to consider that the product is "finished".

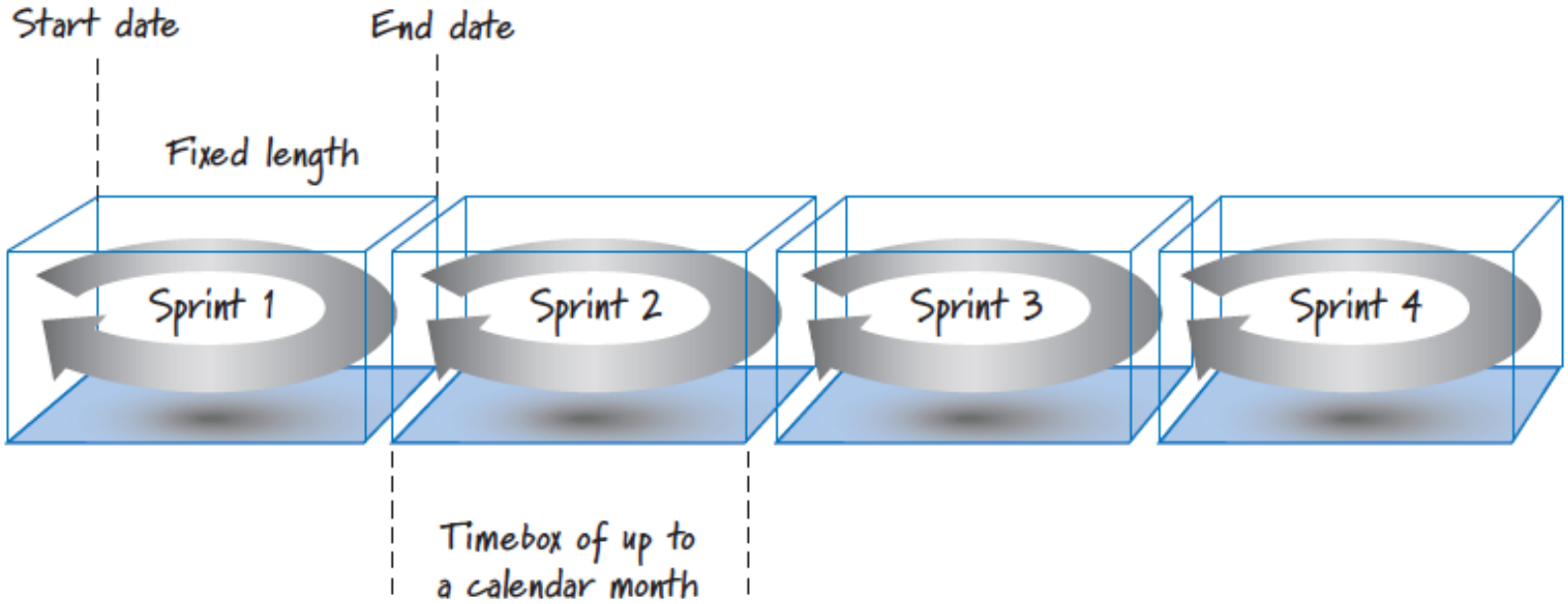
SCRUM – PRACTICES



ACTIVITIES – SPRINT (1)

A Sprint is a time slot (time box) of one month or less during which an increment of the final product is created and "Finished", but also usable and potentially deployable.

It is convenient that the duration of the Sprints is consistent throughout the development effort.



ACTIVITIES – SPRINT (2)

During the Sprint:

- No changes are made that may affect the Sprint goal.
- The scope can be clarified and renegotiated between the Product Owner and the Development Team as they learn more.
- The quality objectives cannot be changed.

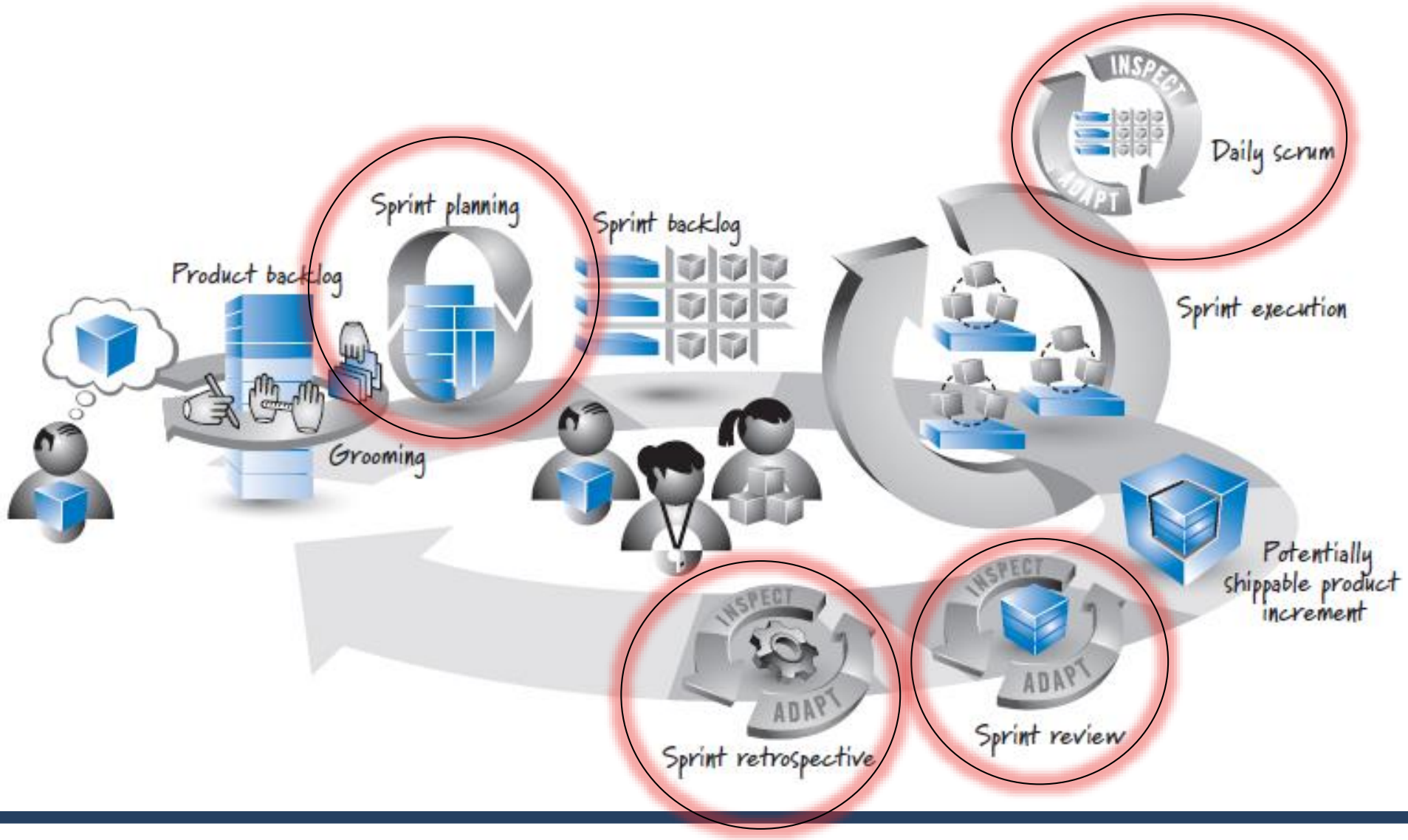
ACTIVITIES – SPRINT (3)

Each Sprint has a definition of what is going to be built, a design, and a flexible plan that will guide the work to be done and the resulting product.

Sprints are limited to one calendar month. Otherwise the definition of what is being built could change, the complexity could rise and the risk could increase.

A Sprint can be cancelled before the end date, but only the Product Owner has the authority to cancel the Sprint, although it can do so under the influence of the interested parties, the development team or the Scrum Master.

ACTIVITIES - MEETINGS



ACTIVITIES – SPRINT PLANNING

GDAŃSK UNIVERSITY OF TECHNOLOGY

It is a working day (no more than 8 hours) meeting conducted prior to the start of each Sprint.

It serves to discuss and determine the work and the objectives that must be covered with the iteration.

This plan is created through the collaborative work of the complete team

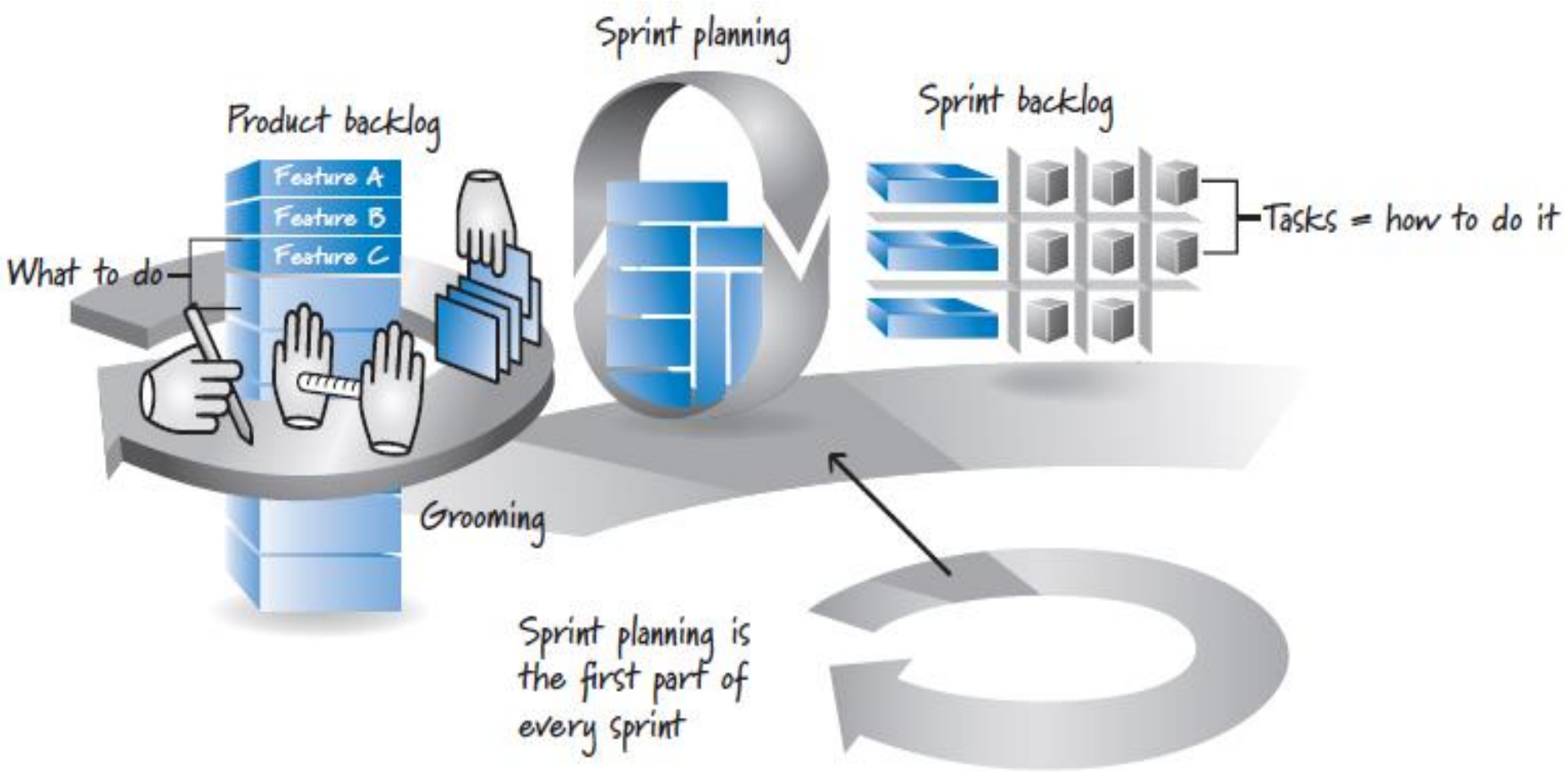
The inputs include: a) product backlog; b) last product increment; c) projected capacity of the team for the Sprint; and d) past performance of the team.

The output is the Sprint Backlog or list of tasks to be performed.

The meeting should provide responses for the following questions:

- ✓ What can we deliver in the increment resulting from the Sprint?
- ✓ How we will do the work necessary to deliver the increment ?

ACTIVITIES – SPRINT PLANNING



ACTIVITIES – DAILY SCRUM

It is a brief daily meeting (15 minutes), only by the development team, to review the progress of each task, and the work planned for the day.

During the meeting, each member of the team explains:

- ✓ What did he/she do yesterday that helped the team achieve the Sprint goal?
- ✓ What will he/she do today to help the team achieve the Sprint goal?
- ✓ What impediments he/she sees that could prevent the team or the person to achieve the Sprint goal?

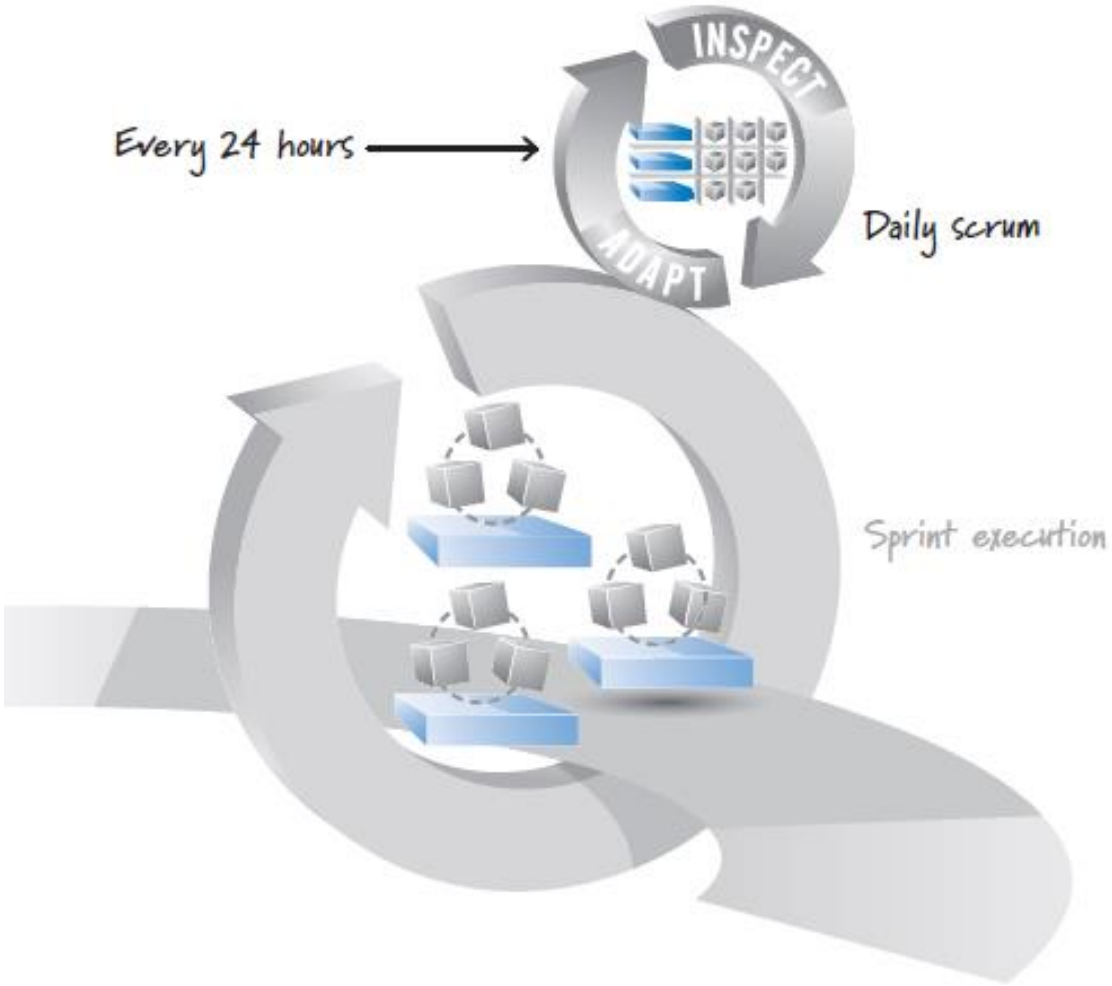
ACTIVITIES – DAILY SCRUM

The Daily Scrum meetings contribute to:

- improve communication
- eliminate the need to hold other meetings
- identify and eliminate impediments related to software development
- highlight and to promote rapid decision making
- improve the level of knowledge of the team.

The Daily Scrum constitutes a key meeting for inspection and adaptation.

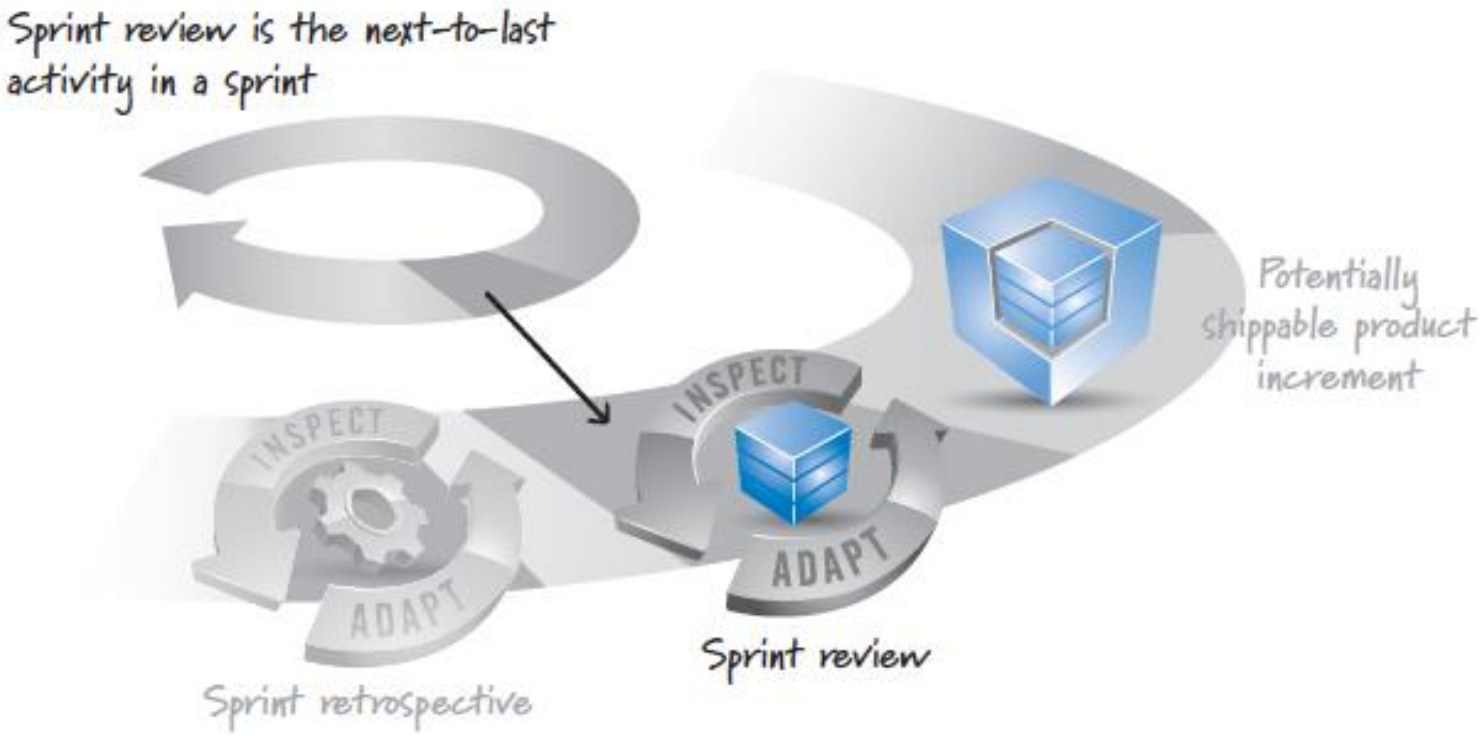
ACTIVITIES – DAILY SCRUM



ACTIVITIES – SPRINT REVIEW (1)

A Sprint Review meeting is conducted at the end of each Sprint to inspect the increment and adapt the Product Backlog, if necessary.

The meeting lasts for at most four hours – A duration of two hours is advisable.



ACTIVITIES – SPRINT REVIEW (2)

GDAŃSK UNIVERSITY OF TECHNOLOGY

The Sprint Review features include:

- ✓ The attendees include the team members and key stakeholders invited by the Product Owner.
- ✓ The Product Owner explains which elements of the Product Backlog have been “Finished" and which ones have not; and discusses the likely completion dates for the project, based on the progress made to date (if necessary);
- ✓ The team shows the work that has been “finished" and answers questions about the increment. They also explain what went well during the Sprint, what problems appeared and how those problems were resolve.
- ✓ The entire group collaborates on what to do next, so that the Sprint Review provides valuable input information for subsequent Sprint Planning Meetings.

ACTIVITIES – SPRINT RETROSPECTIVE

GDAŃSK UNIVERSITY OF TECHNOLOGY

The team reflects on the way in which they did their work and the events that happened in the Sprint that has just concluded to improve their practices.

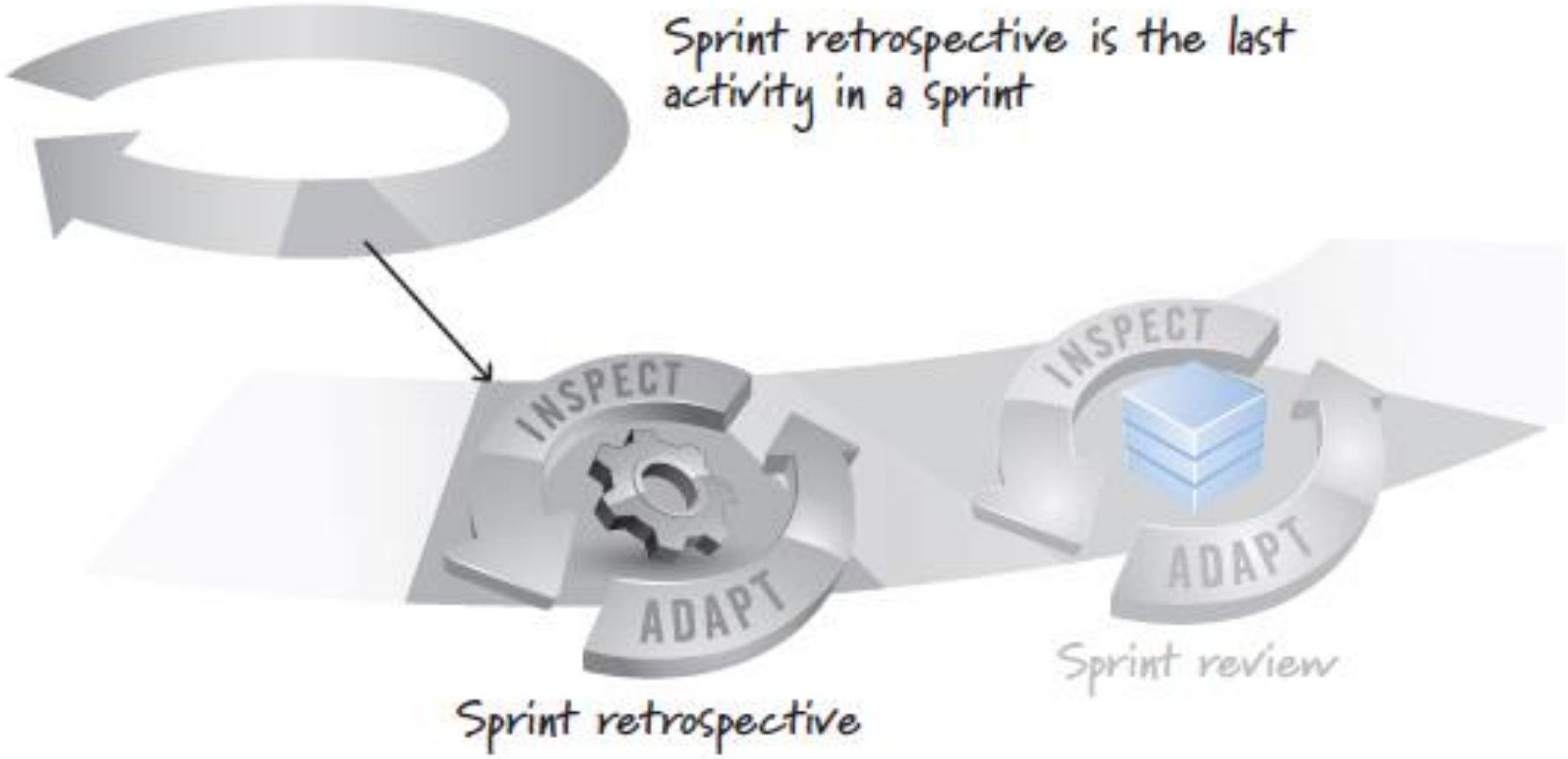
It is an opportunity for the team to inspect itself and create an improvement plan that will be addressed during the next Sprint.

While the review meeting is intended to review the product (the "what"), the retrospective focuses on the process (the "how").

The meeting is expected to last from 30 minutes to one hour.

It is the heart of continuous improvement and emerging practices.

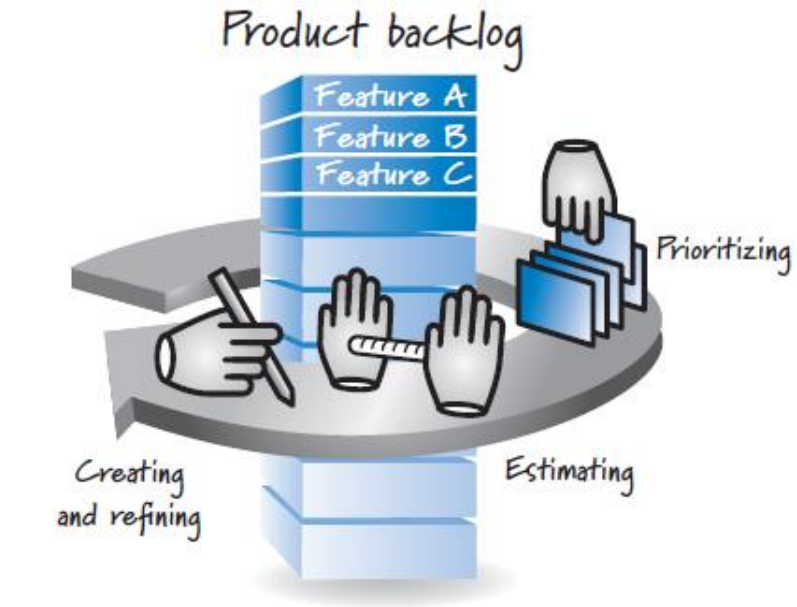
ACTIVITIES – SPRINT RETROSPECTIVE



ACTIVITIES – PRODUCT BACKLOG GROOMING

The Product Owner maintains the Product Backlog updated:

- adjusting priorities based on business value
- adding new items
- eliminating old items



Product backlog grooming

COMPANIES USING SCRUM

The following companies use Scrum	
Microsoft	Nokia
Yahoo	Ericsson
Google	BBC
Philips	Telefónica
Siemmens	...

SCRUM IS USED FOR...

GDAŃSK UNIVERSITY OF TECHNOLOGY

Scrum is used for the following type of projects	
Commercial software	Websites
Internal projects	Mobile apps
Financial applications	Network switching applications
Fixed-price projects	Software applications certified by ISO
Embedded systems	Satellite control software
Software games	...

Ref: Mike Cohn, “An Introduction to Scrum”

GROUP DISCUSSION

- Form groups
- Search for Scrum good practices
- Summarize one good practice per group and identify the source

Possible references include:
http://www.scrumsense.com/wp-content/uploads/2009/12/DoBetterScrum-v2.pdf
http://lookforwardconsulting.com/wp-content/uploads/2011/01/ScrumObservations.pdf

AIM AND AGENDA

AIM

To present and compare project management methodologies.

AGENDA

1	PROCESS	What is software development process and what are agile methods?
2	SCRUM	What are main features of SCRUM?
3	RUP	What are main features of RUP?
4	SUMMARY	What was covered in this section?

UNIFIED PROCESS - OVERVIEW

RUP provides a rigorous approach to assigning tasks and responsibilities in the scope of a software development project.

RUP aims at ensuring the production of high-quality software that meets the needs of its end-users, within a predictable schedule and budget.

Main features:

1) iterative and incremental

2) use case-driven

3) architecture-centric

1 – ITERATIVE AND INCREMENTAL – HOW?

GDAŃSK UNIVERSITY OF TECHNOLOGY

The design process is based on iterations that address different aspects of the design process.

The iterations evolve into the final system (incremental aspect).

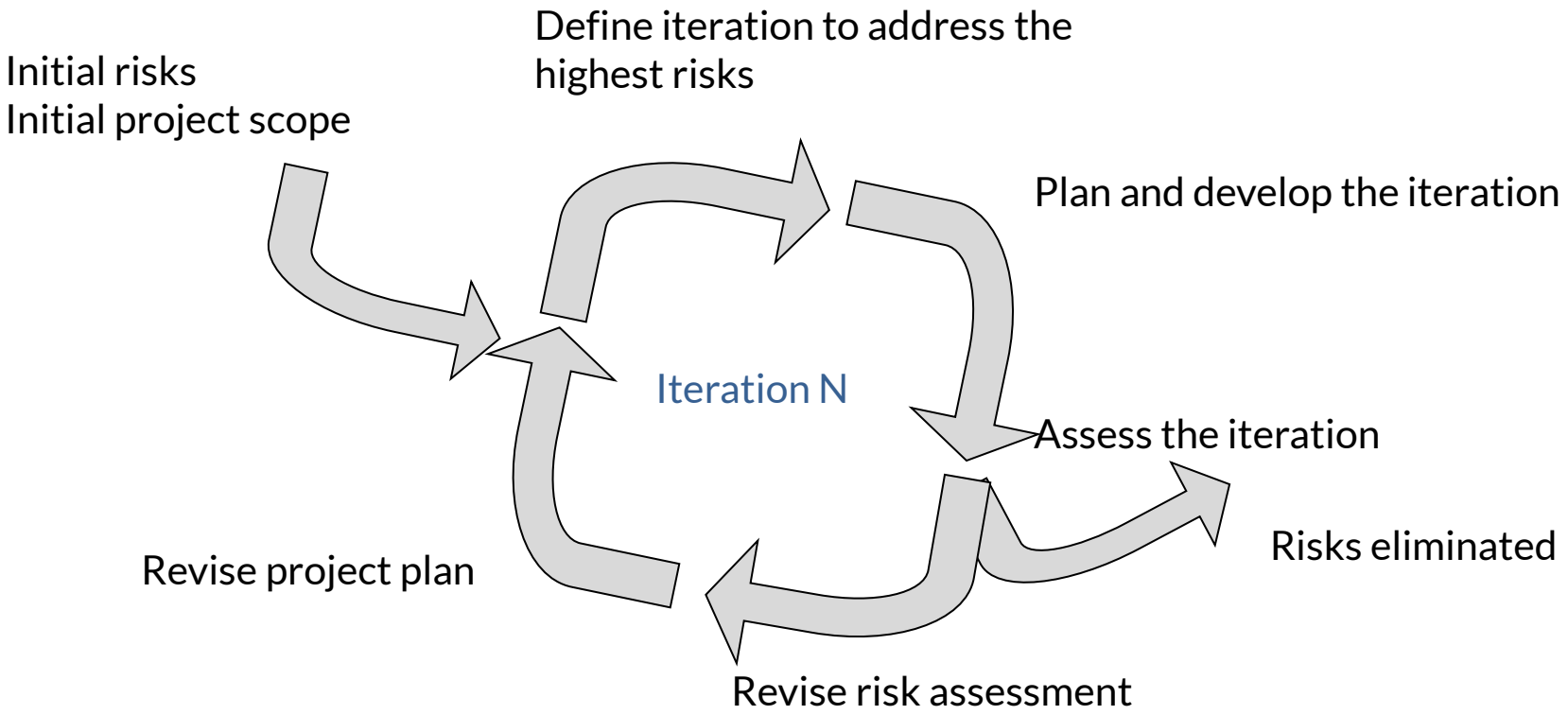
The process does not try to complete the whole design task in one shot.

How to do it?

1) plan a little
2) specify, design and implement a little
3) integrate, test and run
4) obtain feedback before next iteration

1 - ITERATIVE AND INCREMENTAL – RISKS

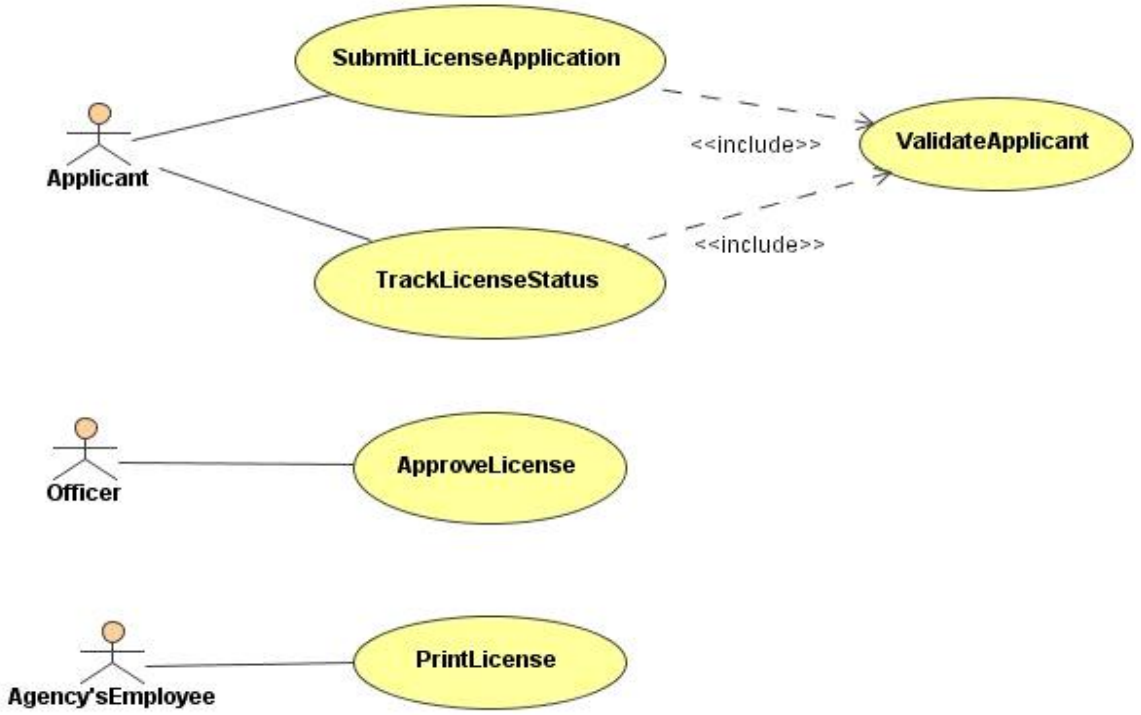
Technical risks are assessed and prioritized early and are revised during each iteration.



(USE CASE)

A use case is a description of a sequence of actions that a system performs to deliver an observable result to a particular actor, it is realized by collaboration.

Use cases serve to structure the behavioral elements in a model



2 – USE CASE-DRIVEN

Use cases are used for:

- identifying users and their requirements
- aiding in the creation and validation of the architecture
- helping produce definitions of test cases and procedures
- directing the planning of iterations
- driving the creation of user documentation
- directing the deployment of the system
- synchronizing the content of different models
- driving traceability throughout models

3 – ARCHITECTURE-CENTRIC

Problem - with the iterative and incremental approach different development activities are done concurrently

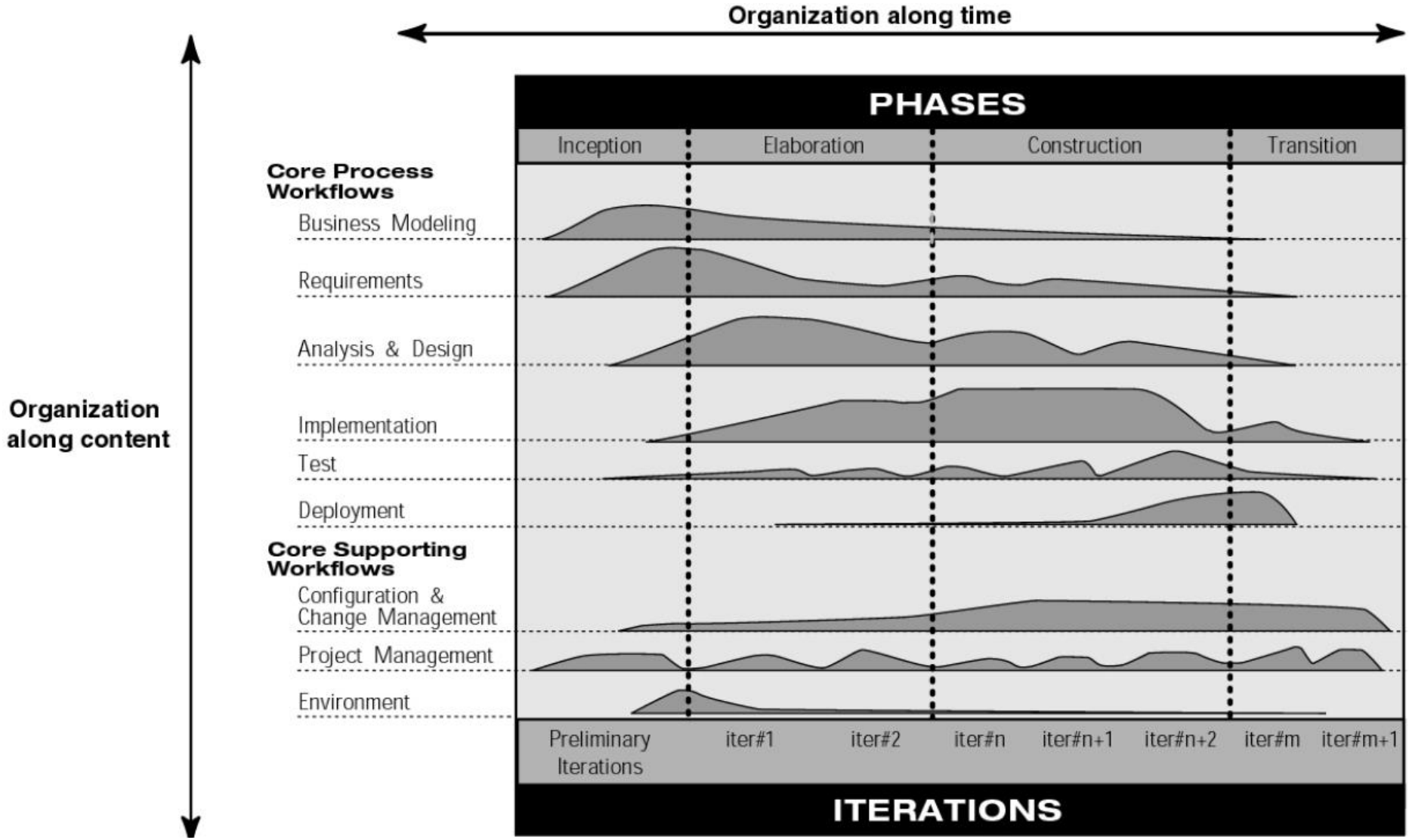
Solution - the system's architecture ensures that all parts fit together

"An architecture is the skeleton on which the muscles (functionality) and skin (user-interface) of the system will be hung".

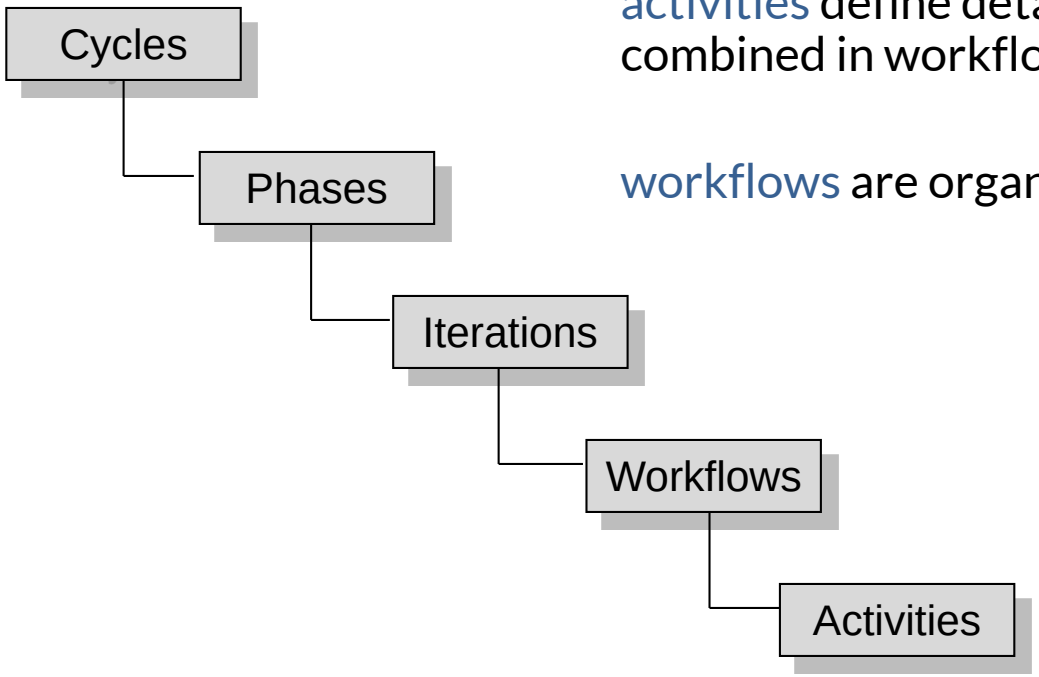
UNIFIED PROCESS - STRUCTURE

Two dimensions:	
Time	division of the life cycle into phases and iterations
Process components	production of a specific set of artifacts with well-defined activities called workflows

THE DEVELOPMENT PROCESS



BUILDING BLOCKS



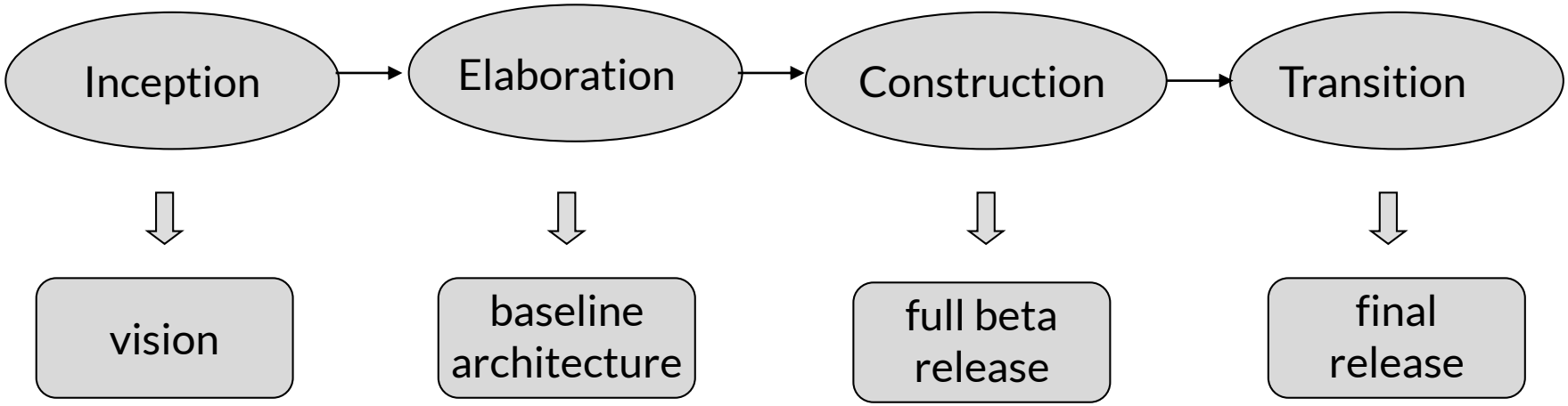
activities define detailed work and are combined in workflows

workflows are organized into iterations

- each iteration identifies some aspect of the system and is organized into phases
- phases can be grouped into cycles
- cycles focus on the generation of successive releases

LIFE CYCLE PHASES

Phases and major deliverables:



INCEPTION

Focuses on the generation of the business case that involves:

- identification of core uses cases
- definition of the actual scope
- identification of risky difficult parts of the system

The main objectives are:

- to prove the feasibility of the system to be built
- to determine the complexity involved in order to provide reasonable estimates

Outputs:

1) the vision of the system
2) very simplified use case model
3) tentative architecture
4) risks identified
5) plan for the elaboration phase

ELABORATION

Involves:

- understanding how the requirements are translated into the details of the system
- producing the baseline architecture
- capturing the majority of the use cases
- exploring further the risks identified earlier and identifying the most significant
- specifying any non-functional (quality) requirements specially those related to reliability and performance

Outputs:

1) the system's architecture
2) detailed use case model
3) set of plans for the construction phase

CONSTRUCTION

Involves:

- completing the analysis of the system
- performing the majority of the design and the implementation

Outputs:

1) implemented software product as a full beta release. It may contain some defects
2) associated models

An important aspect for the success of this phase is to monitor the critical aspects of the project, specially significant risks.

TRANSITION

Involves:

- deployment of the beta system
- monitoring user feedback and handling any modifications or updates required

Output:

1) the formal release of the software

RUP - WORKFLOWS 1

GDAŃSK UNIVERSITY OF TECHNOLOGY

1) Requirements - focuses on the activities which allow to identify functional and non-functional requirements

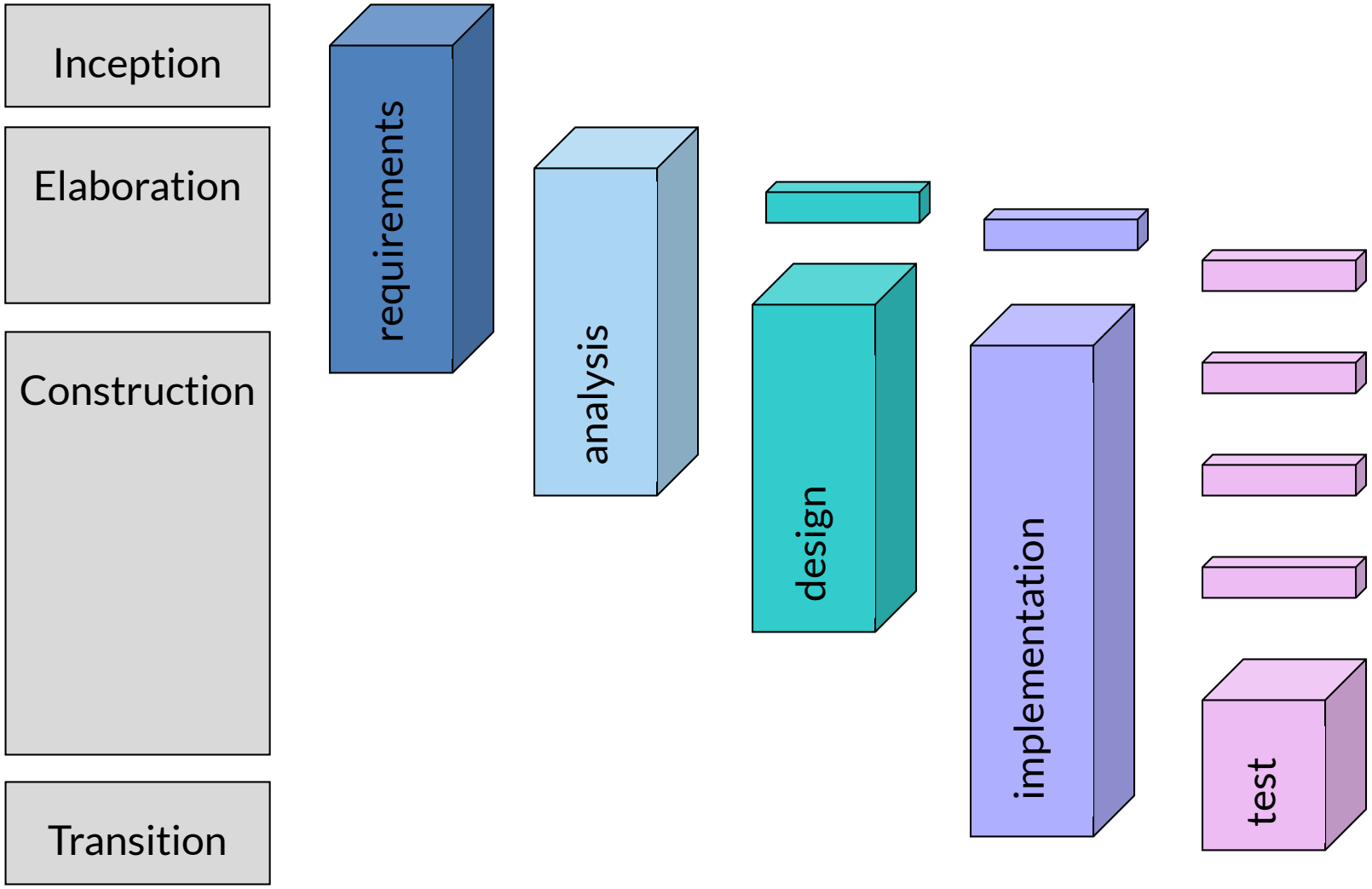
2) Analysis - restructures the requirements identified in terms of the software to be built

3) Design - produces a detailed design

4) Implementation - represents the coding of the design in a programming language, and the compilation, packaging, deployment and documentation of the software

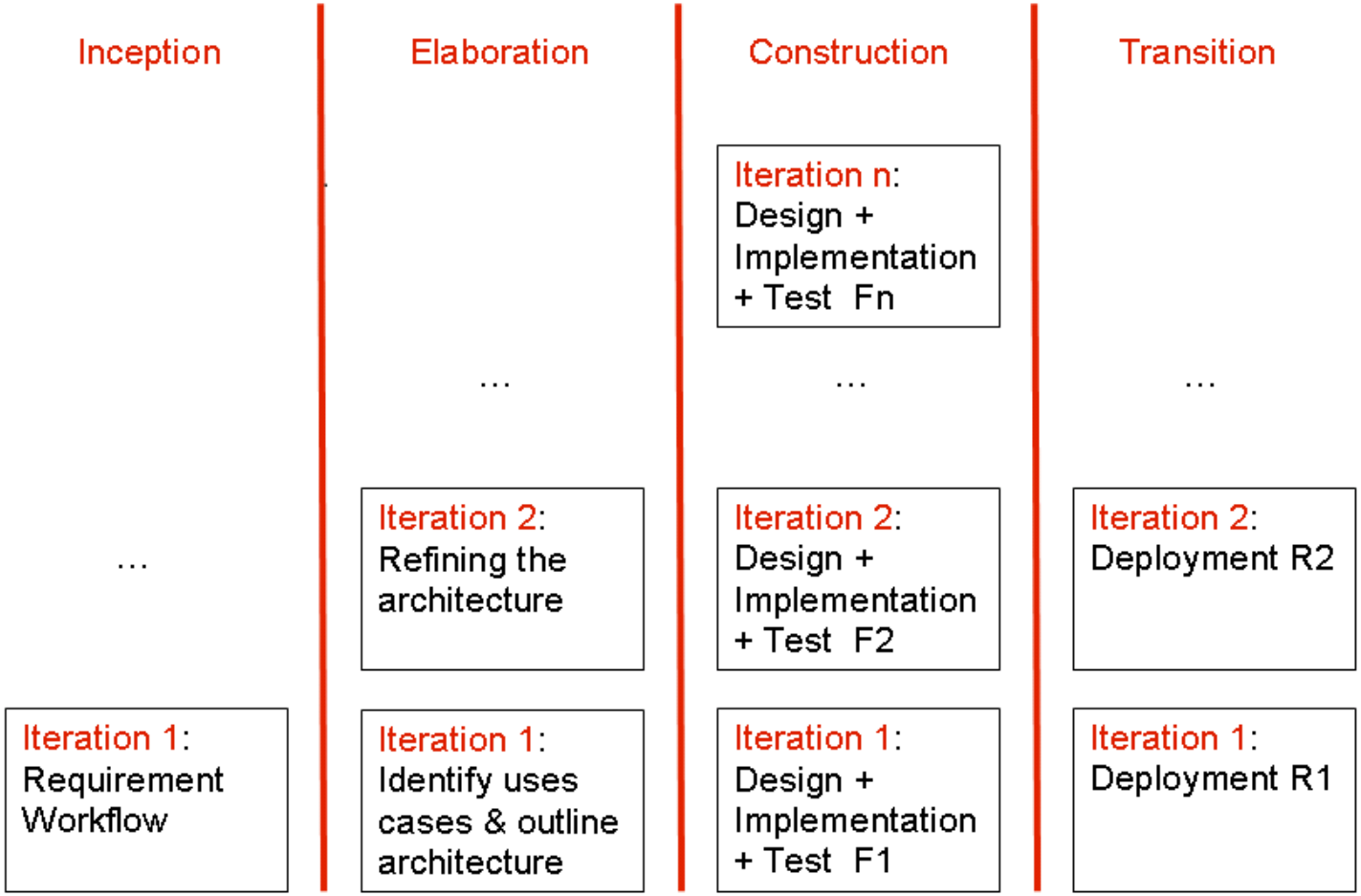
5) Test - describes the activities to be carried out for testing

WORKFLOWS AND PHASES



PHASES AND ITERATIONS

The iterations of workflows occur one or more times during a phase.

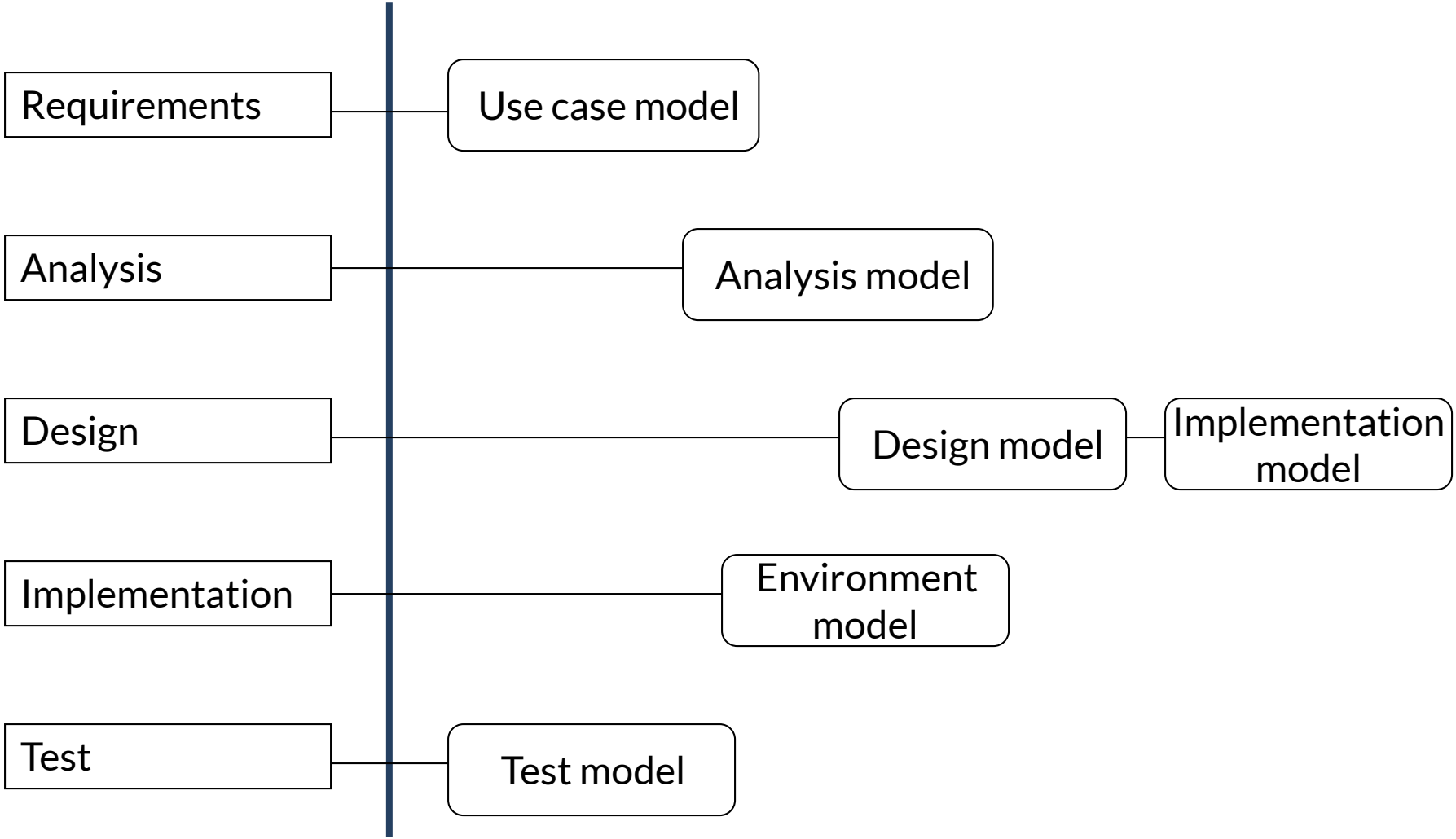


WORKFLOWS AND ACTIVITIES

GDAŃSK UNIVERSITY OF TECHNOLOGY

Workflows	Activities
Requirements	Find actors and use cases, prioritize use cases, refine use cases, prototype user interface, structure the use case model
Analysis	Analyze possible architectures, analyze use cases, explore classes, find packages
Design	Design the architecture, trace use cases, refine and design classes, design packages
Implementation	Implement the architecture, implement classes and interfaces, implement subsystems, perform unit testing, integrate systems
Test	Plan and design tests, implement tests, perform integration and system tests, evaluate tests

WORKFLOWS AND MODELS



COMPANIES USING RUP

The following companies use RUP	
Telecommunications	Ericsson, Alcatel
Transportation, airspace defense	Lockheed-Martin, British Aerospace
Manufacturing	Xerox, Volvo, Intel
Finances	Visa, Merrill Lynch, Schwab
System integrators	CAP Gemini Ernst & Young, Oracle, Deloitte & Touche

GROUP DISCUSSION

- Form groups
- Search for RUP good practices
- Summarize one good practice per group and identify the source

Possible references include:

https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf

https://www.cscjournals.org/manuscript/Journals/IJSE/Volume5/Issue2/IJSE-142.pdf

AIM AND AGENDA

AIM

To present and compare project management methodologies.

AGENDA

1	PROCESS	What is software development process and what are agile methods?
2	SCRUM	What are main features of SCRUM?
3	RUP	What are main features of RUP?
4	SUMMARY	What was covered in this section?

SUMMARY – SW DEVELOPMENT METHODS

GDAŃSK UNIVERSITY OF TECHNOLOGY

A software development methodology (method) defines who is doing what, when to do it

how to reach a certain goal, and the inputs and outputs for each activity of a software development process.

Agile methodologies are a kind of software development methodology applying the values and principles of the Agile Manifesto.

Agile methodologies value:

1) people and interactions over processes and tools
2) software over documentation
3) collaboration with client over contract negotiation
4) response to change over following a plan

SUMMARY – SCRUM (1)

SCRUM is an agile methodology for developing software.

SCRUM pillars are visibility, inspection, adaptability

SCRUM practices are:

- incremental development
- review of iterations
- self-organized and multi-functional team
- collaboration
- prioritization criteria defined by the client

SUMMARY – SCRUM (2)

SCRUM Roles – Product Owner, Scrum Master, Team.

SCRUM Artefacts – Product Backlog, Sprint Backlog, Increment

SCRUM Activities:

- Sprint
- Sprint Planning
- Daliy Scrum
- Sprint Execution
- Sprint Review
- Sprint Retrospective
- Product backlog grooming

RUP (1)

RUP is a software development methodology (considered a product and a process).

RUP is:

1) iterative and incremental
2) use case-driven
3) architecture-centric

RUP the process can be modeled into two dimensions:

1) Time - division of the life cycle into phases and iterations
2) Process components - a specific set of artifacts with well-defined activities called workflows

RUP (2)

RUP Phases – Inception, Elaboration, Construction, Transition

RUP Workflows:

- Core Process Workflows – Business Modeling. Requirements, Analysis and Design, Implementation, Test, Deployment
- Core Supporting Workflows – Configuration and Change Management, Project Management, Environment

(ADDITIONAL) GROUP DISCUSSION

- Form groups
- Identify similarities and differences between SCRUM and RUP

BIBLIOGRAPHY – SCRUM

GDAŃSK UNIVERSITY OF TECHNOLOGY

REFERENCES

- *Essential Scrum - A Practical Guide to the Most Popular Agile Process,* Kenneth S. Rubin, 2014.
  Chapter 2 is available here:
  http://agileforall.com/wp-content/uploads/2014/07/Essential_Scrum_Chapter_2-.pdf

- *Succeeding wit Agile. Software Development using Scrum*, Mike Cohn, 2010.

- *Scrum Handbook*, Jeff Sutherland
  https://www.researchgate.net/publication/301685699_Jeff_Sutherland's_Scrum_Handbook

BIBLIOGRAPHY – RUP

REFERENCES

- Rational Unified Process - Best Practices for Software Development Teams Rational Software White Paper, TP026B, Rev 11/01 https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf

- The Rational Unified Process, Chapter 2, http://catalogue.pearsoned.co.uk/samplechapter/0321197704.pdf

BIBLIOGRAPHY

ADDITIONAL READING

- "Manifesto for Agile Software Development," https://agilemanifesto.org/
- Mendes Calo, K., Estevez, E., Fillottrani, P., "A Quantitative Framework for the Evaluation of Agile Methodologies", Journal of Computer Science & Technology, vol.10 (2), pp.68-73, http://journal.info.unlp.edu.ar/JCST/article/view/729/258
- West, D., "Planning a Project with the Rational Unified Process", http://www.nyu.edu/classes/jcf/g22.2440-001_sp09/handouts/PlanningProjWithRUP.pdf

ACKNOLEDGEMENTS

CONTENT OF THE SCRUM SECTION

- Most of the slides of the Scrum section are based on the slides prepared by the academic team of Laboratory of Applied Informatics, Universidad Nacional de Río Negro. Special thanks to Luis Vivas, Mauricio Tassara and Marcelo Petroff

- The figures about Scrum have been copied from Kenneth Rubin, Essential Scrum – *A Practical Guide to the Most Popular Agile Process*.

Many thanks!

Elsa Estevez
ecestevez@gmail.com