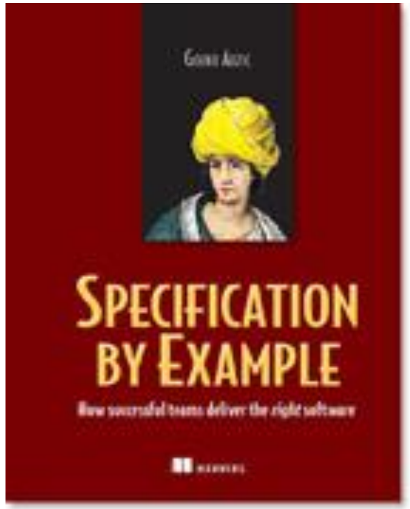
Specification By Example for Educational Purposes

Isabelle BLASQUEZ
Limoges University
France

Hervé Leblanc
IRIT – Toulouse
France

Specification By Example



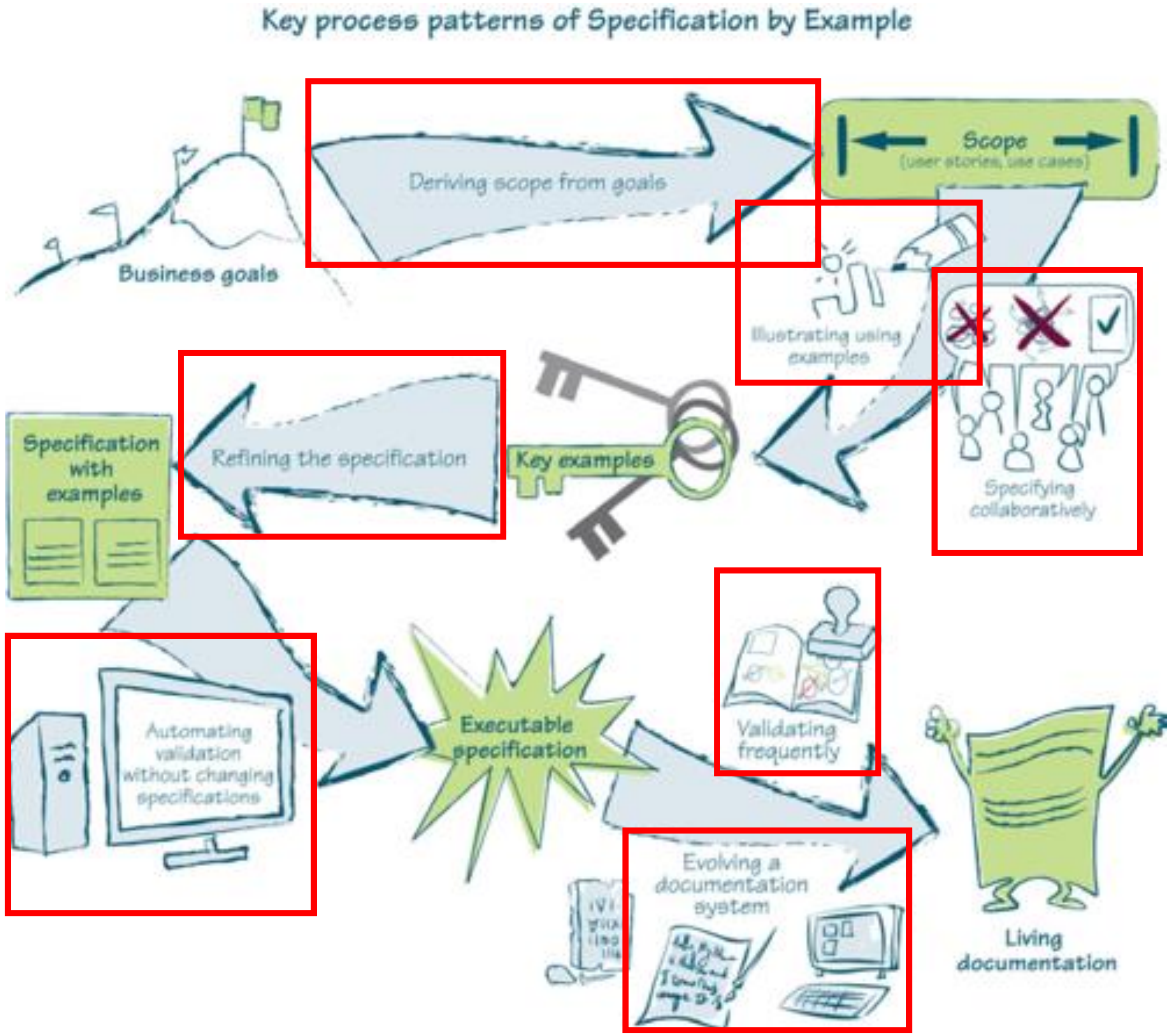
Specification by example (SBE) is a **collaborative approach** for specifying requirements and business-oriented functional tests.

It is consists on **seven process patterns.**
(identified by studying over 50 software projects)

*It is a guideline that assures delivery team and business skateholders that the **product software build it's right for its purpose***

It produces **living, reliable documentation.**
It defines **expectations clearly**
and makes ***validation efficient*.**

The 7 Process Pattern of Specification By Example


Key process patterns of Specification by Example
Business goals
Deriving scope from goals
Scope
(user stories, use cases)
Illustrating using examples
Specifying collaboratively
Key examples
Refining the specification
Specification with examples
Automating validation without changing specifications
Executable specification
Validating frequently
Evolving a documentation system
Living documentation

1. Deriving scope from goals
2. Specifying collaboratively
3. Illustrating requirements using example
4. Refining the specification
5. Automating validation without changing specification
6. Validating frequently
7. Evolving a documentation system

What about in Educational Context ?

A fact : Boredom, inattention, discouragement, poorly results on tests,…
when mismatches between **learning and teaching styles**

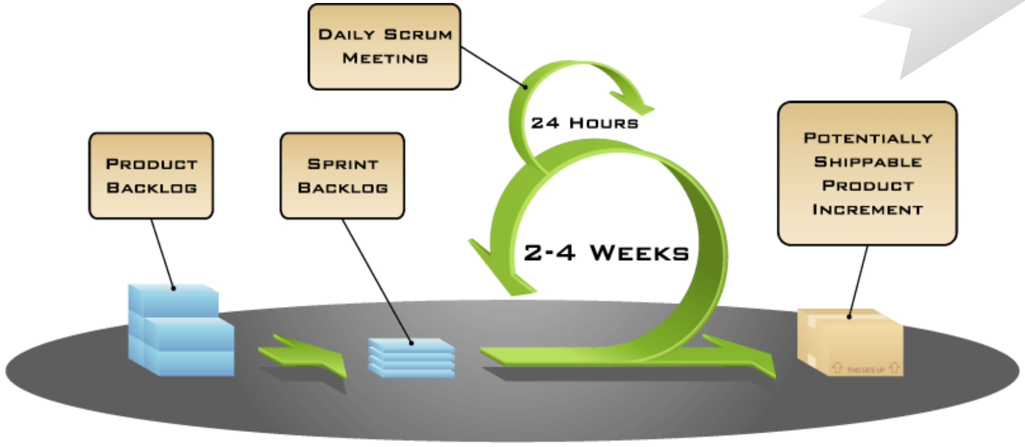
To improve this alignment,
why not trying a connection between Software Engineering & Educational ?

Could not a **course** be considered as a **product** ?

Could not a **Syllabus** be considered as a **Requirements ?**

Credit: Thinkstock

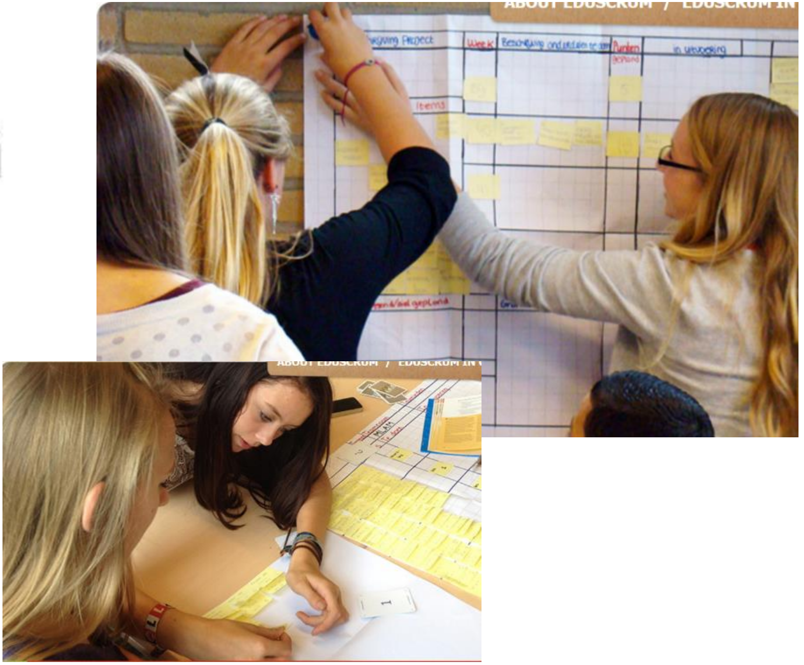
Some transpositions already exist …
… inspired by Scrum



Copyright © 2005, Mountain Goat Software

Scrum :
iterative and incremental
agile software development framework
for managing product development

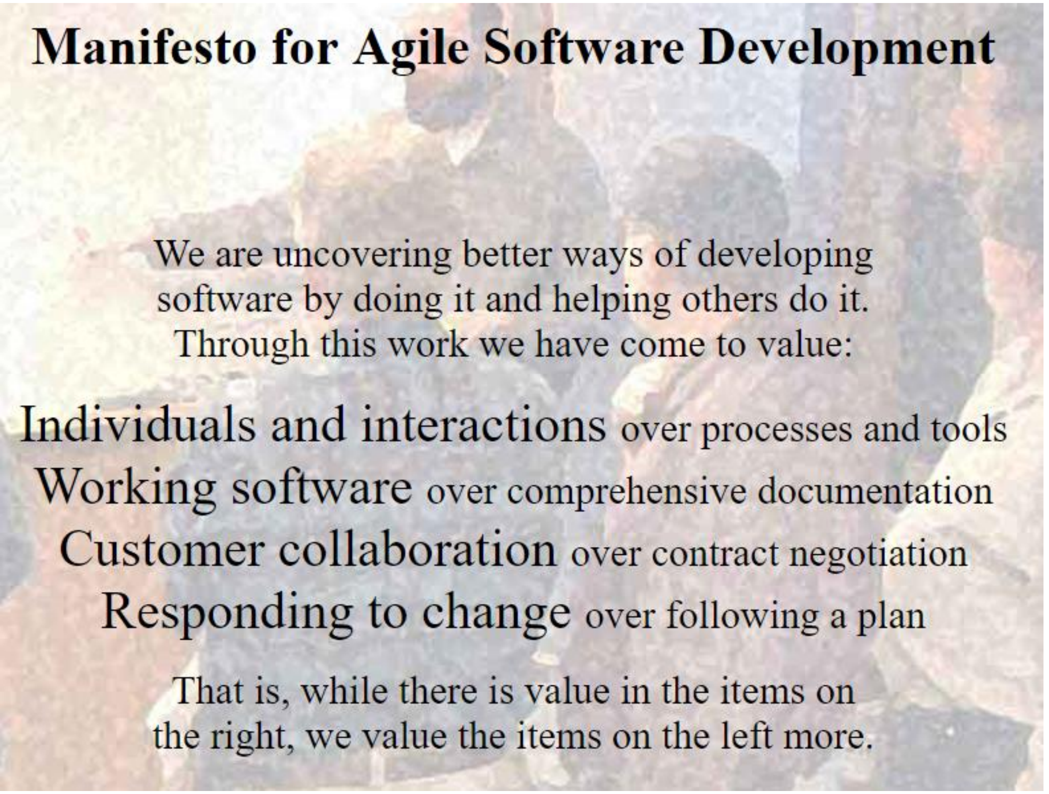
http://eduscrum.nl/en/



http://approchealpes.info/

(private joke in french :
Agi**L**es pour la **P**édagogie dans l'**E**nseignement **S**upérieur)

Also (individual) experiments of *Agile Pedagogy* … inspired directly by agile values and principles



Manifesto for Agile Software Development

We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on
the right, we value the items on the left more.

In France : https://pedagogieagile.com/
Individual experiments of a teacher
in a Junior high school

… As in spain

FPdGi Fundació Fundación Princesa de Girona

SEMINARIO

PEDAGOGÍAS ÁGILES
DEL EMPRENDIMIENTO

FORMACIÓN EN COMPETENCIA
EMPRENDEDORA PARA DOCENTES

2016-2017

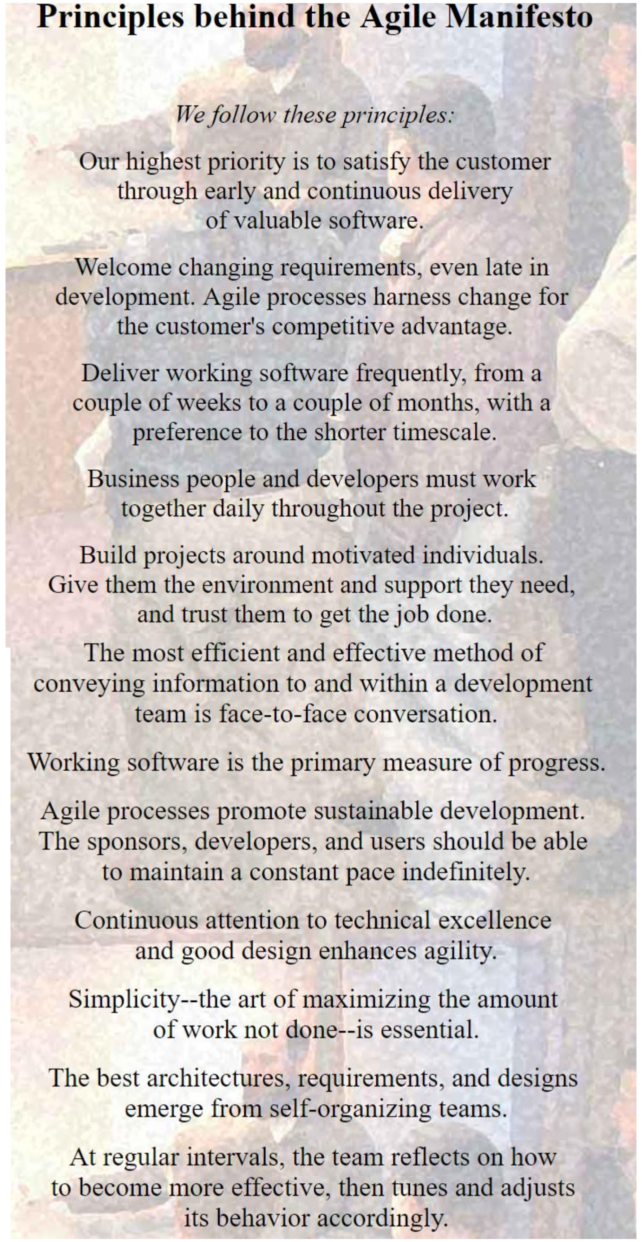
APRENDER
DE EMPRENDER
CÓMO EDUCAR EL TALENTO EMPRENDEDOR

Fundación TRILEMA

https://es.fpdgi.org/upload/projecte/1329.pdf

… Up to suggest an agile manifesto school

Agile manifesto :
From a traditional plan-driven paradigm
to a **value-driven paradigm**



Principles behind the Agile Manifesto

We follow these principles:

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Business people and developers must work together daily throughout the project.

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Working software is the primary measure of progress.

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

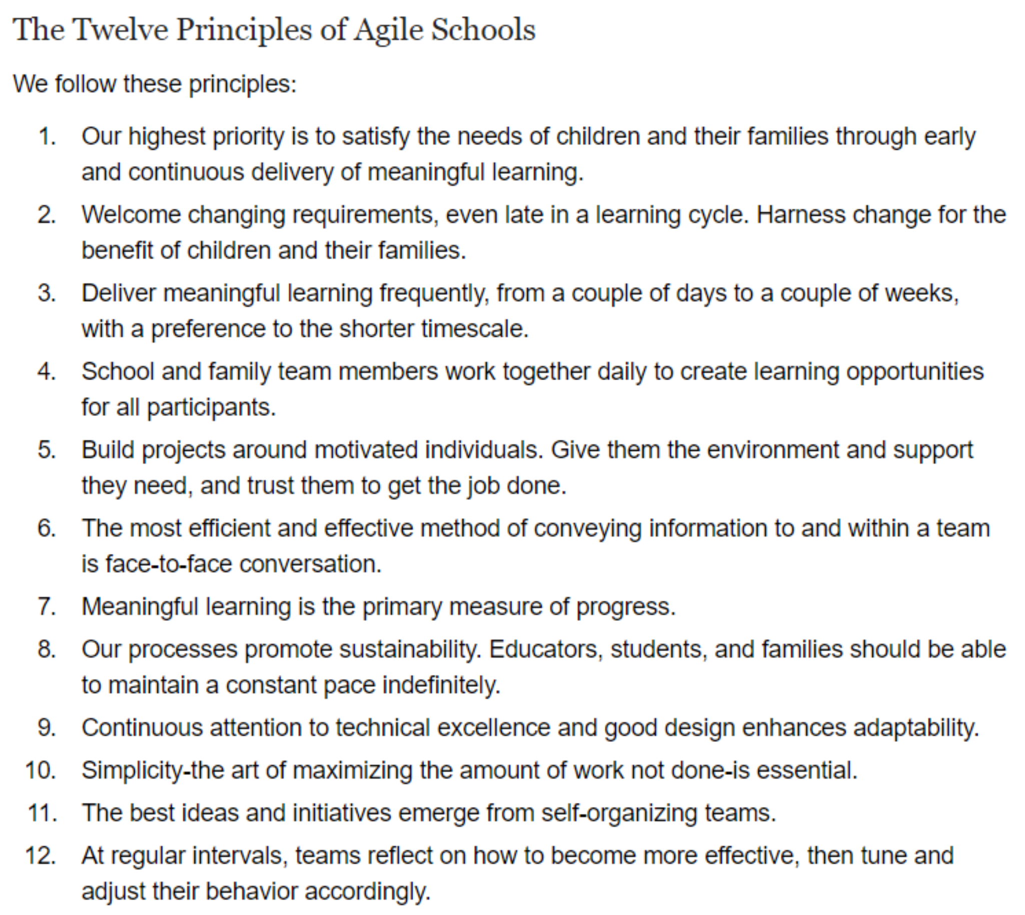
Continuous attention to technical excellence and good design enhances agility.

Simplicity--the art of maximizing the amount of work not done--is essential.

The best architectures, requirements, and designs emerge from self-organizing teams.

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Agile school manifesto :
From traditional teaching-approaches to
new **learning approaches.**



The Twelve Principles of Agile Schools

We follow these principles:

1. Our highest priority is to satisfy the needs of children and their families through early and continuous delivery of meaningful learning.
2. Welcome changing requirements, even late in a learning cycle. Harness change for the benefit of children and their families.
3. Deliver meaningful learning frequently, from a couple of days to a couple of weeks, with a preference to the shorter timescale.
4. School and family team members work together daily to create learning opportunities for all participants.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a team is face-to-face conversation.
7. Meaningful learning is the primary measure of progress.
8. Our processes promote sustainability. Educators, students, and families should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances adaptability.
10. Simplicity-the art of maximizing the amount of work not done-is essential.
11. The best ideas and initiatives emerge from self-organizing teams.
12. At regular intervals, teams reflect on how to become more effective, then tune and adjust their behavior accordingly.

https://www.infoq.com/articles/agile-schools-education

But only transposition around delivery process,
… No transposition around the design ….

Could not a **Course Designer** be considered as a **Product Manager ?**

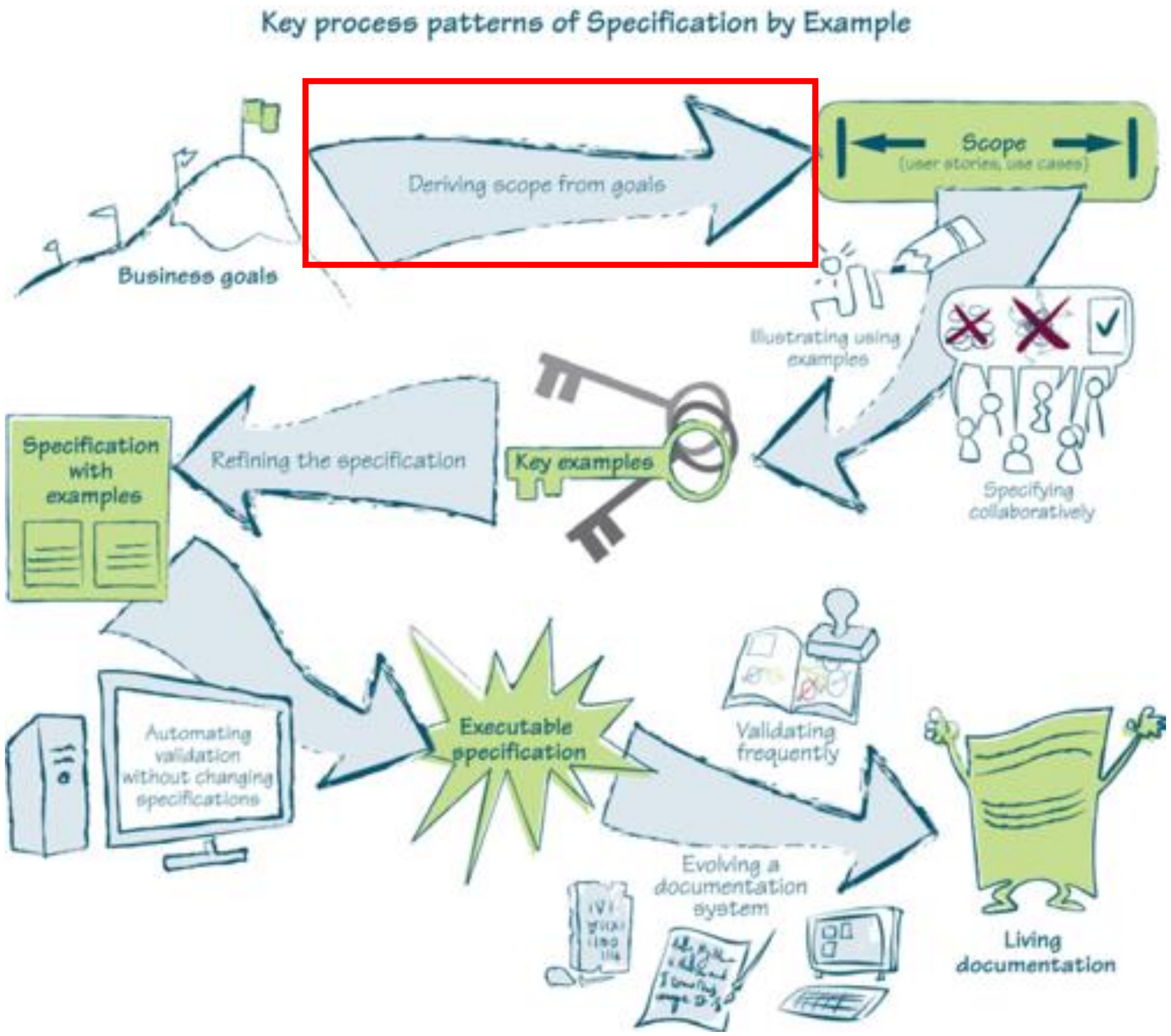
Why not transposing each process pattern of Specification by Example in Educational context to improve the design of a course ?

Transposition of each pattern in 3 steps

→ **Overview** of the **original pattern**

→ Corresponding **pattern in teaching-domain**

→ **Example** of the teaching-domain pattern
on the software development processes course

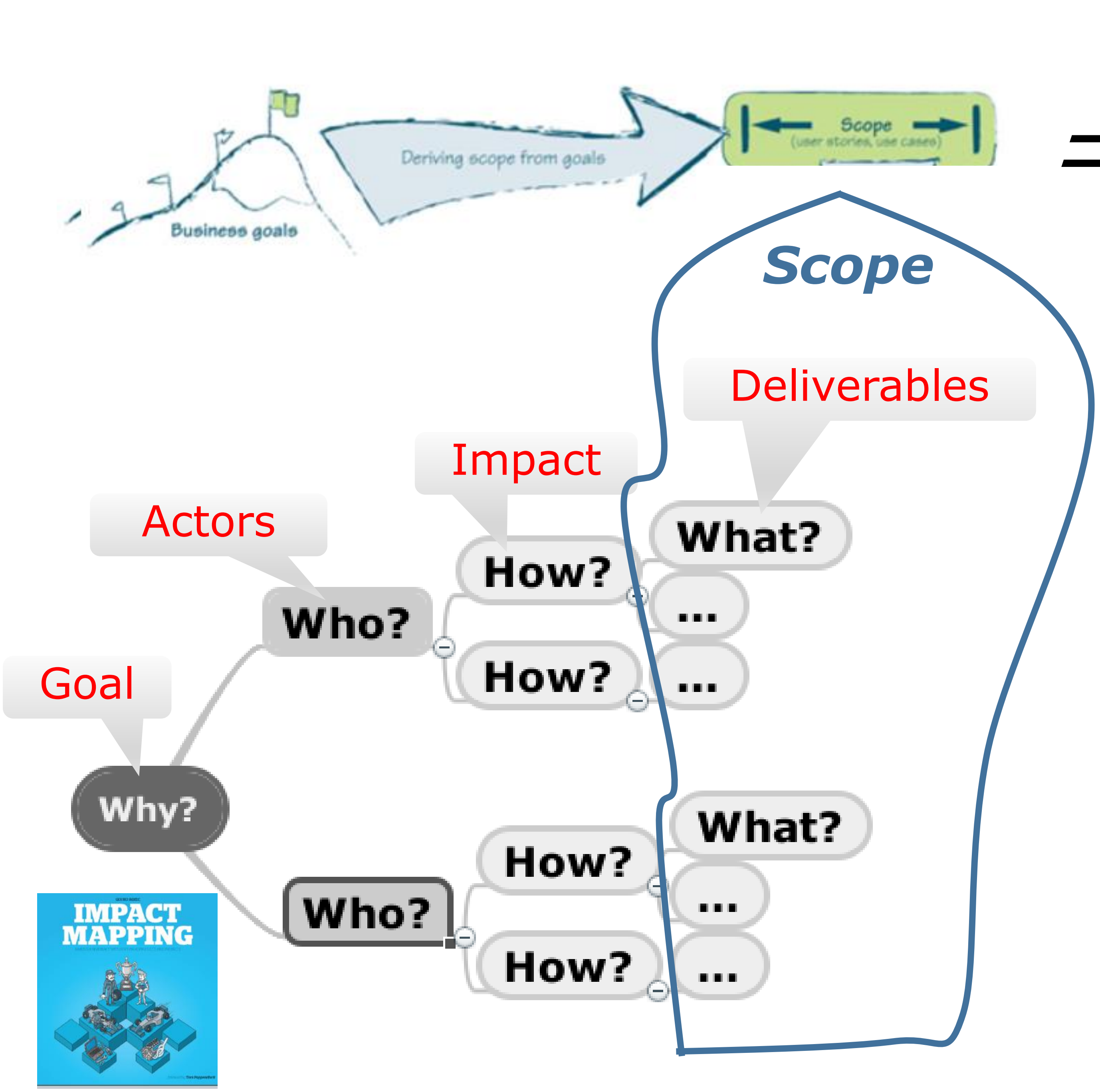

Key process patterns of Specification by Example
Business goals
Deriving scope from goals
Scope
(user stories, use cases)
Illustrating using examples
Specifying collaboratively
Key examples
Refining the specification
Specification with examples
Automating validation without changing specifications
Executable specification
Validating frequently
Evolving a documentation system
Living documentation

Pattern 1 :

Deriving scope from goals

From *why* to *what*

Overview & corresponding pattern



Business goals

Deriving scope from goals

Scope

(user stories, use cases)

Scope

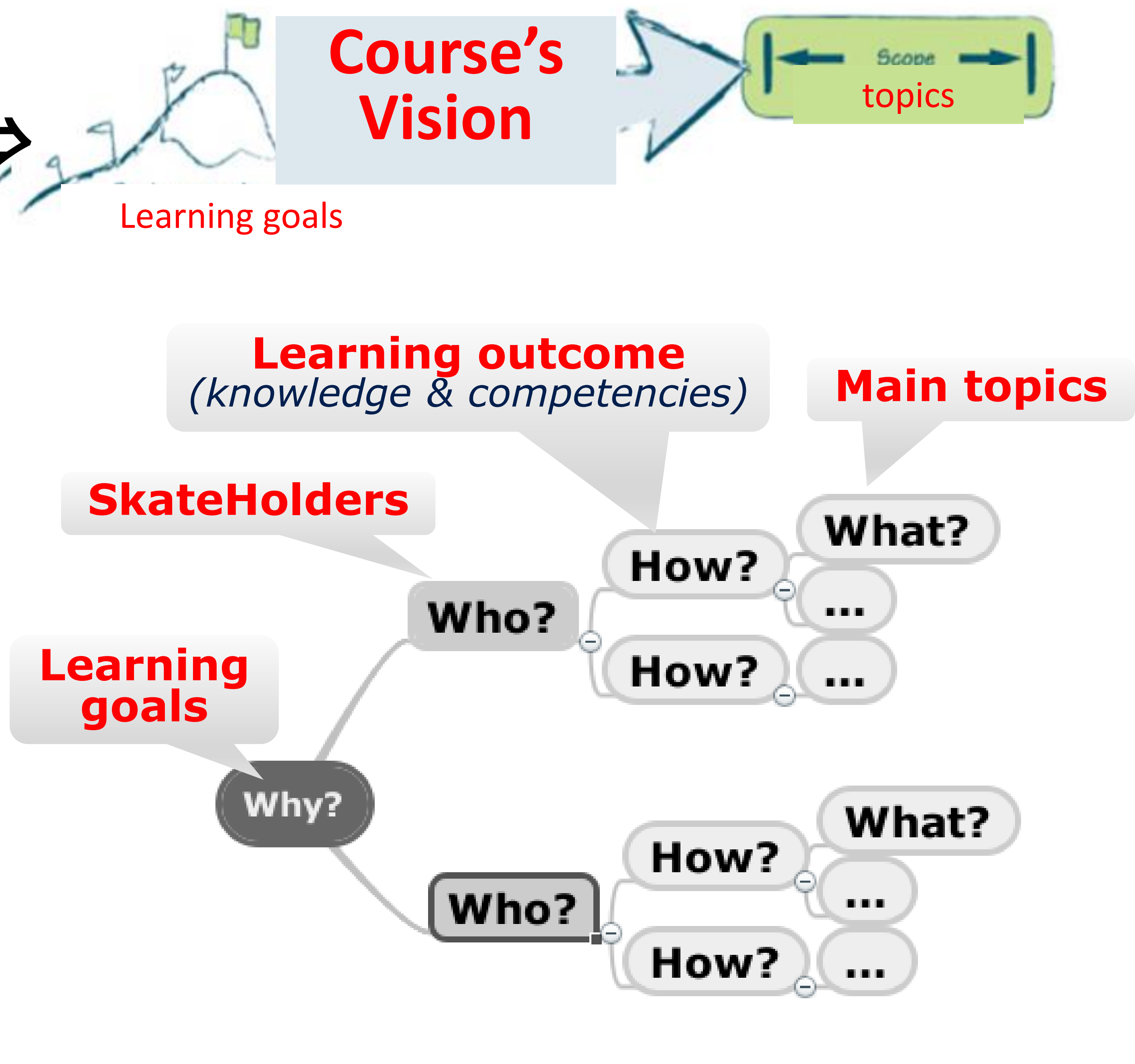
Goal

Actors

Impact

Deliverables

⇒



Course's Vision

Scope

topics

Learning goals

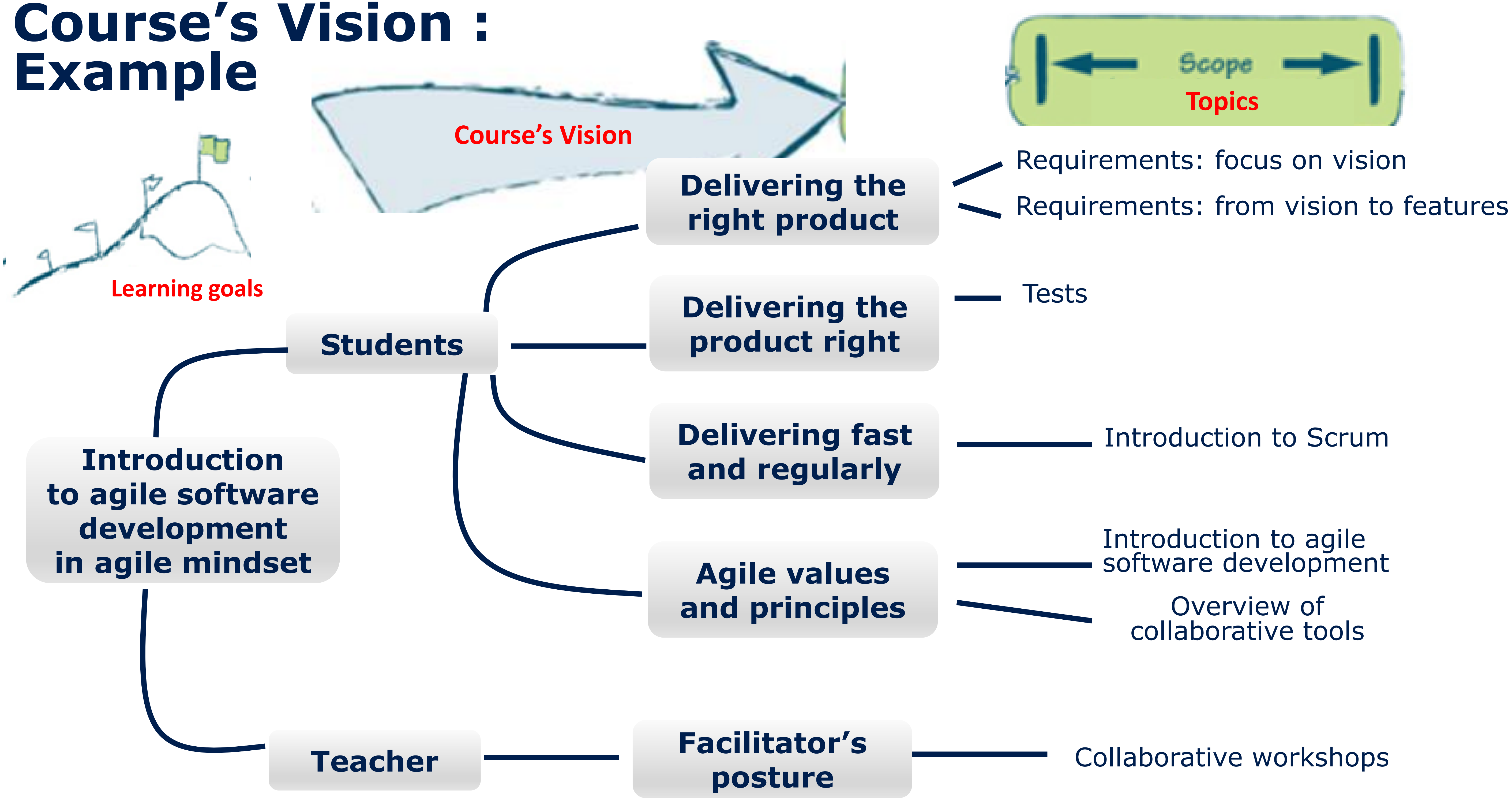
Learning outcome
(knowledge & competencies)

Main topics

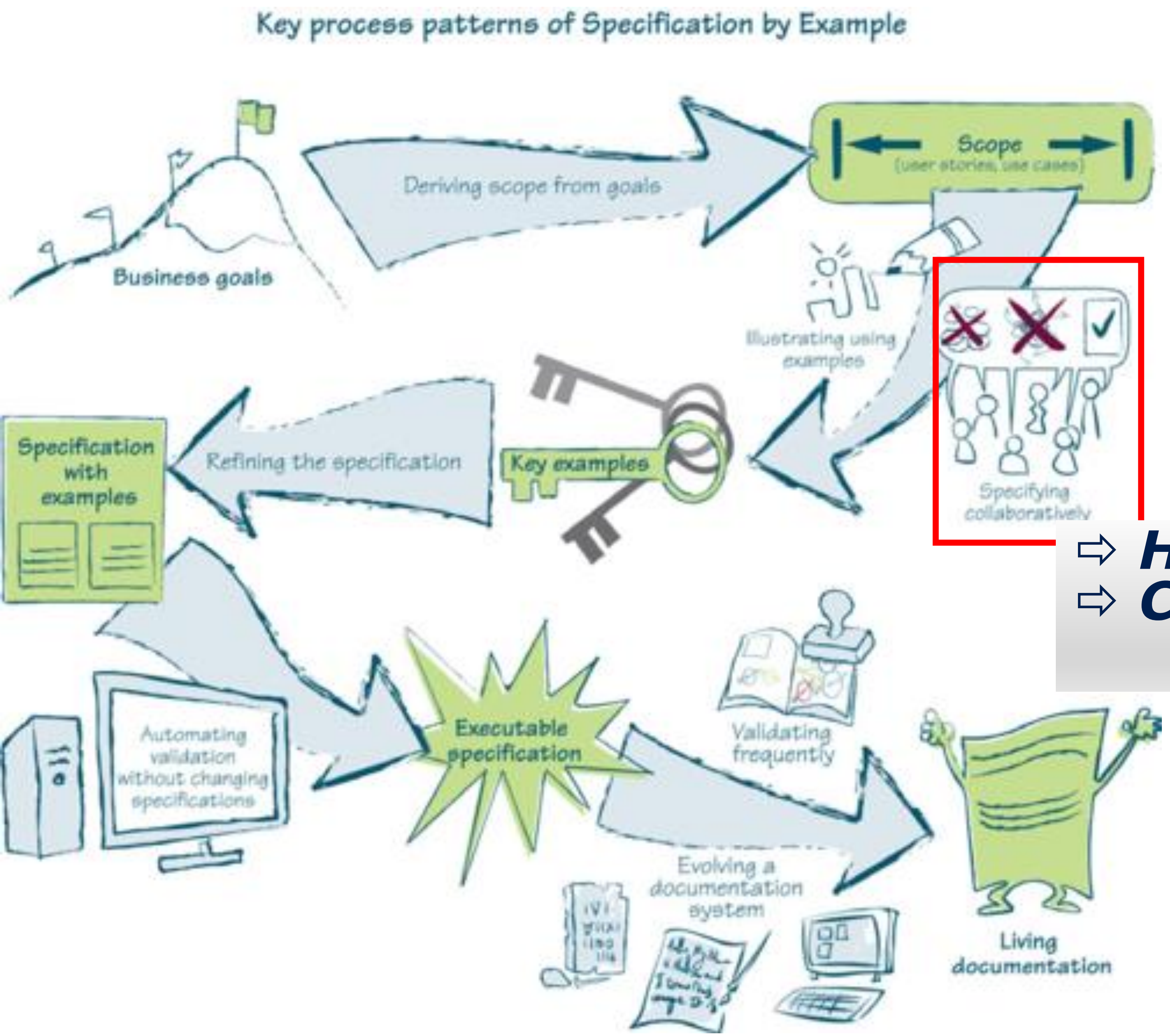
SkateHolders

Learning goals

Course's Vision : Example



- Introduction to agile software development in agile mindset
  - Students
    - Delivering the right product
      - Requirements: focus on vision
      - Requirements: from vision to features
    - Delivering the product right
      - Tests
    - Delivering fast and regularly
      - Introduction to Scrum
    - Agile values and principles
      - Introduction to agile software development
      - Overview of collaborative tools
  - Teacher
    - Facilitator's posture
      - Collaborative workshops

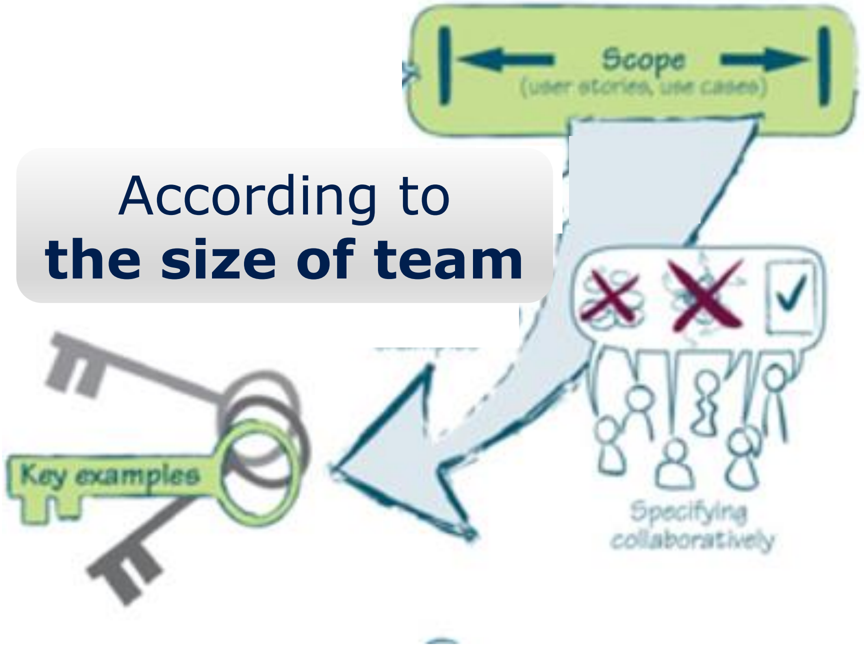

Key process patterns of Specification by Example
Business goals
Deriving scope from goals
Scope
(user stories, use cases)
Illustrating using examples
Specifying collaboratively
Key examples
Refining the specification
Specification with examples
Automating validation without changing specifications
Executable specification
Validating frequently
Evolving a documentation system
Living documentation

Pattern 2 :

Specifying collaboratively

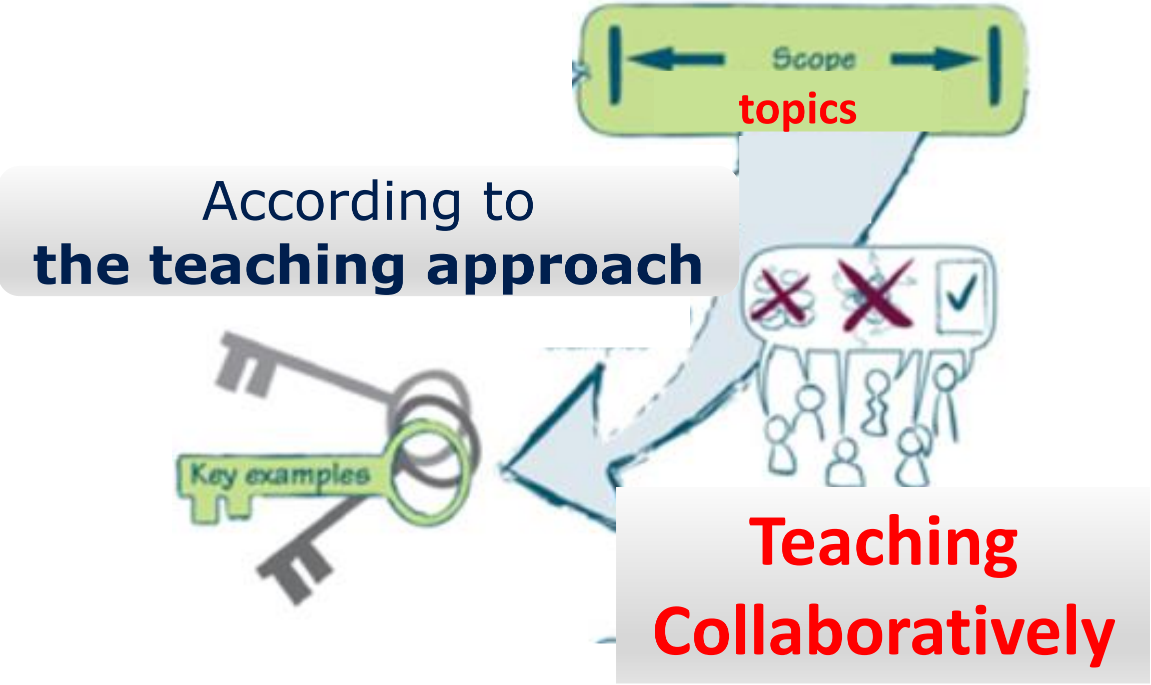
- ⇨ ***Harness the knowledge and the experience***
- ⇨ ***Create a collective ownership of specifications (engagement in the delivery process)***

Overview & corresponding pattern



According to **the size of team**

⇒



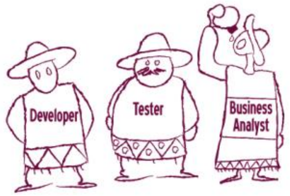
topics

According to **the teaching approach**

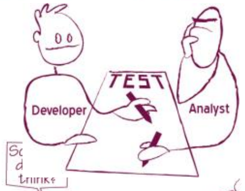
Teaching Collaboratively

All-team workshops

Smaller workshops
(clarify)



Pair-writing
(mature or complete)



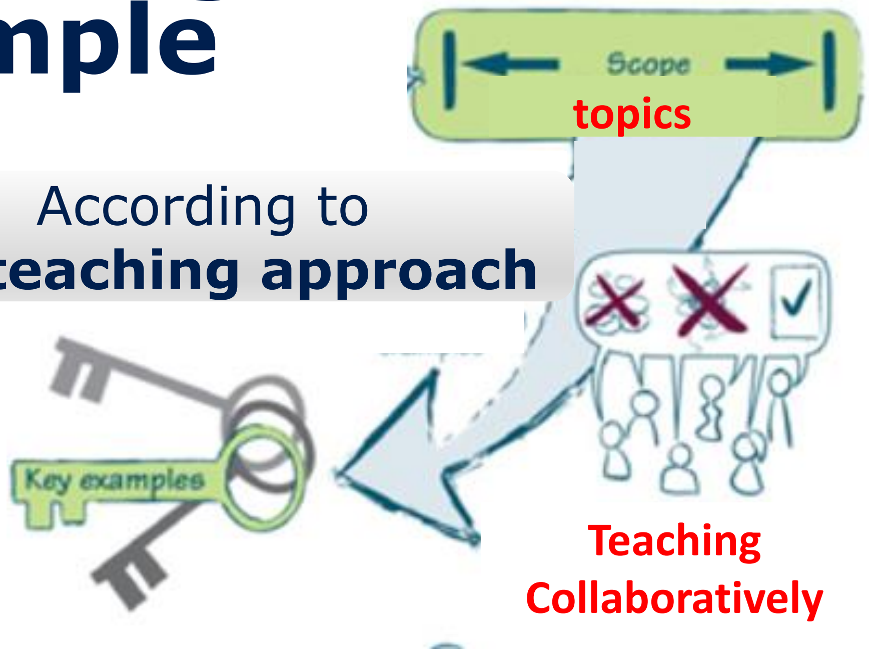
Informal conversations
(several perspectives)

Deductive workshops
(traditional teaching approach)

Inductive workshops
(from particulars to generalities)
→ Problem Based Learning, discovery learning, inquiry learning (constructivism)
→ Active learning (involvement of students) and collaborative learning (groups)

Pictures from : http://www.methodsandtools.com/archive/collaborativespecifications.php

Teaching Collaboratively : Example



topics

According to
the teaching approach

Teaching Collaboratively

Deductive workshops
→ lecture

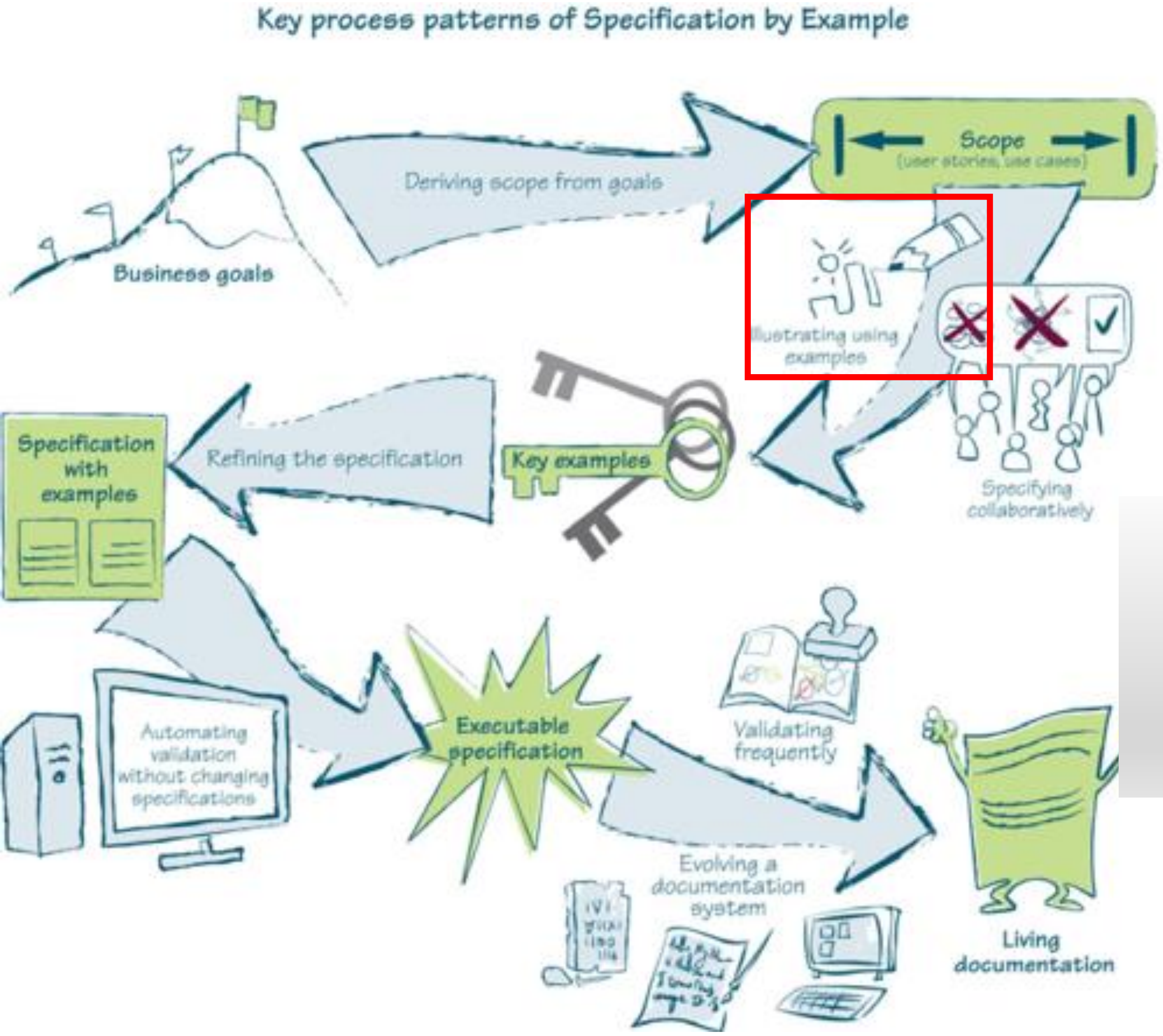
Inductive workshops (mostly)

Autonomous team of 4-6 students who collaborate to achieve the required learning goal

Teacher become **facilitator** who ensure trust behaviors
(demonstrate respect, create transparency, keep commitments, extended trust…)

Assertion time (at the end) :
- **Workshop review :** presentation of work
- **Retrospective :** to reflect on the most significant events to have occurred, examine the lessons learned and take decisions aiming at improvement.

→ Respect the Collective Intelligence process
→ Promote the emergence of Agile Value


Key process patterns of Specification by Example
Business goals
Deriving scope from goals
Scope
(user stories, use cases)
Illustrating using examples
Specifying collaboratively
Key examples
Refining the specification
Specification with examples
Automating validation without changing specifications
Executable specification
Validating frequently
Evolving a documentation system
Living documentation

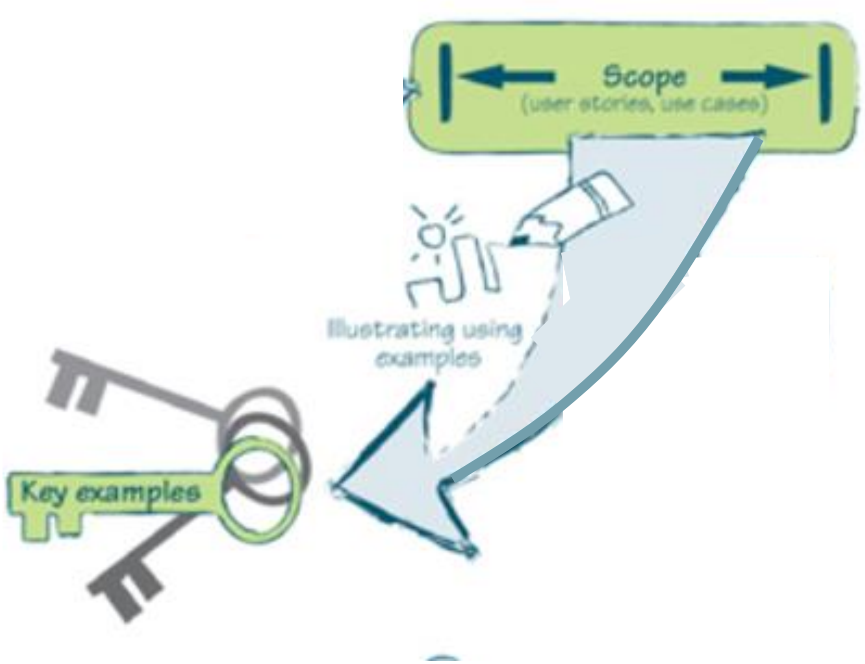
Pattern 3 :

Illustrating requirements using example

Ensure that the delivery teams focus on the right product

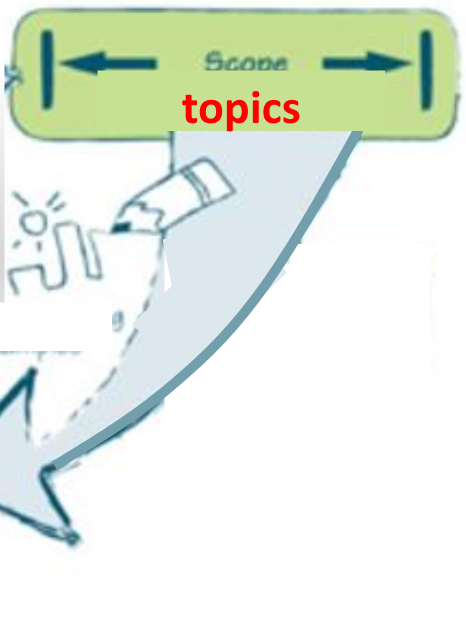
Shared understandings of what the business users expect

Overview & corresponding pattern



⇒

Illustrating with teaching resources (support)



Example

Teaching resource
activity used by a teacher
to engage students
in learning
to achieve required
learning goal

Examples (everyday conversation)
→ clarify meaning
→ concrete less unambigous

Examples (from requirements analysis to testing)
→ small
→ precise
→ realistic
→ easy to understand

Illustrating with teaching resources : Example

Project Based Learning

Student centered approach :
Work in groups on a real world problem,
Guidance provided by the teacher
Resulting products shared with the community

Lecture

To introduce or clarify a concept

Choose the best teaching resources to engage students in learning

Tutorial, Lab

Collaborative workshops

Gamification

process of using-game based mechanics, aesthetics and game thinking to engage people, motivate action, promote learning and solve problem [1]

(Game ⇒) culture happiness ⇒ learning ⇒ productivity [2]

[1] K. Kapp. The Gamification of Learning and Instruction: Game-based Methods and Strategies for Training and Education. Wiley, 2012

[2] D. Mezick. The Culture Game: Tools for the Agile Manager. FreeStanding Press, 2012

Illustrating with teaching resources : Example (in picture)

Lecture

To introduce or clarify a concept

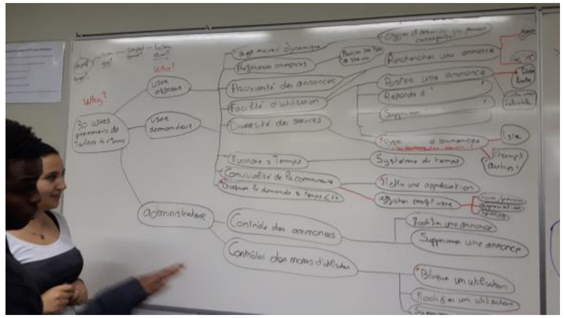
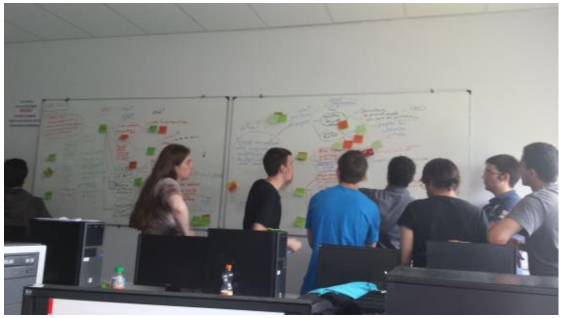
Choose the best teaching resources to engage students in learning

Tutorial, Lab

Project Based Learning

Student centered approach :
Work in groups on a real world problem,
Guidance provided by the teacher
Resulting products shared with the community

Collaborative workshops

Gamification

Serious Game

(to discover a concept)

Innovation Game

(to work)

Happiness to work (photos taken by students during PBL workshops)

Game/Collaborative Workshop ↔
culture happiness ↔ learning ↔ productivity

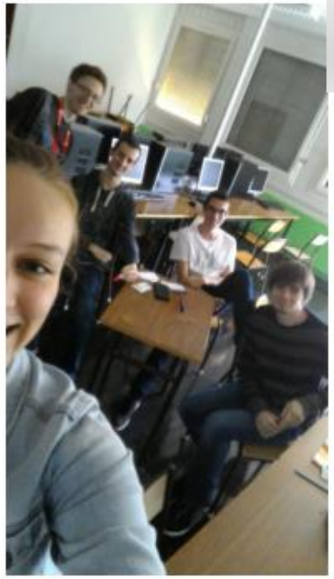


Photo prise lors de l'atelier des User Stories

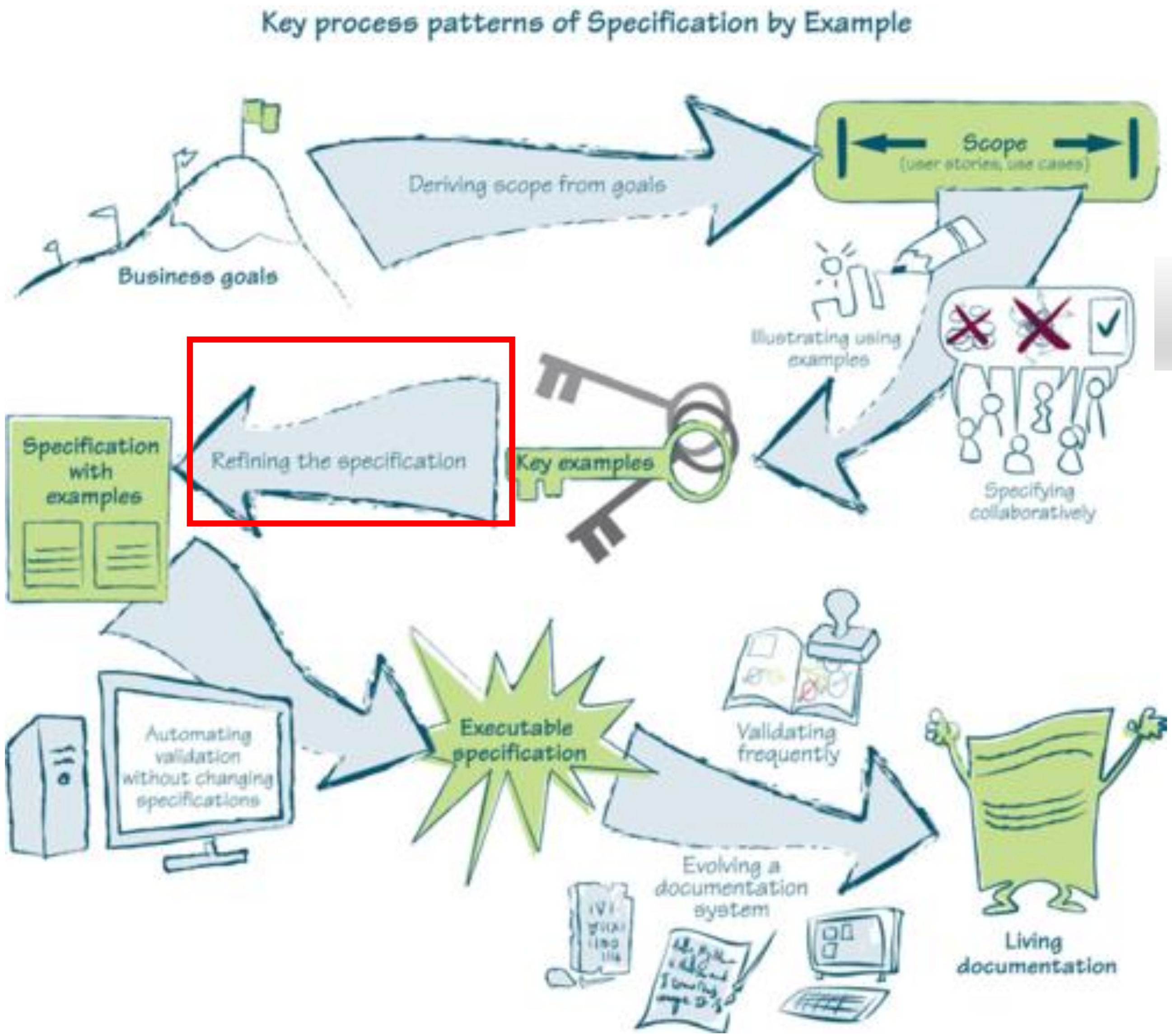
Dessin réalisé lors de l'atelier Impact Mapping, où vous étiez relativement « Speedy ».

Conclusion graphique

Pour illustrer à quel point nous formons une équipe soudée et efficace, nous avons quelques Snapchats qui vous montrent comment nous avons passé le temps lors de nos pauses.

Les BeeGees

Et enfin pour finir, la vidéo que vous attendiez depuis si longtemps : la danse de Pierre, Sébastien, et Conor, reprise et modifiée par d'autres MDI (membres du département informatique) : Cliquez ici pour voir the best conclusion ever.


Key process patterns of Specification by Example
Business goals
Deriving scope from goals
Scope
(user stories, use cases)
Illustrating using examples
Specifying collaboratively
Key examples
Refining the specification
Specification with examples
Automating validation without changing specifications
Executable specification
Validating frequently
Evolving a documentation system
Living documentation

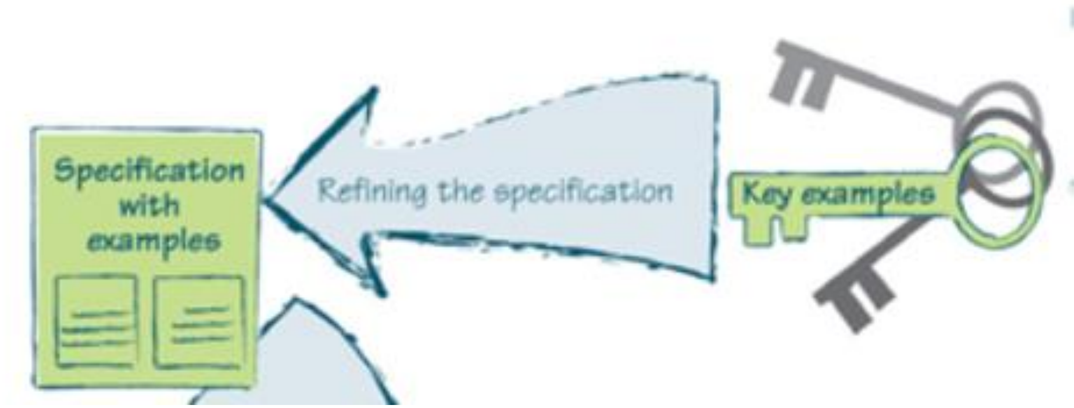
Pattern 4 :

Refining the specification

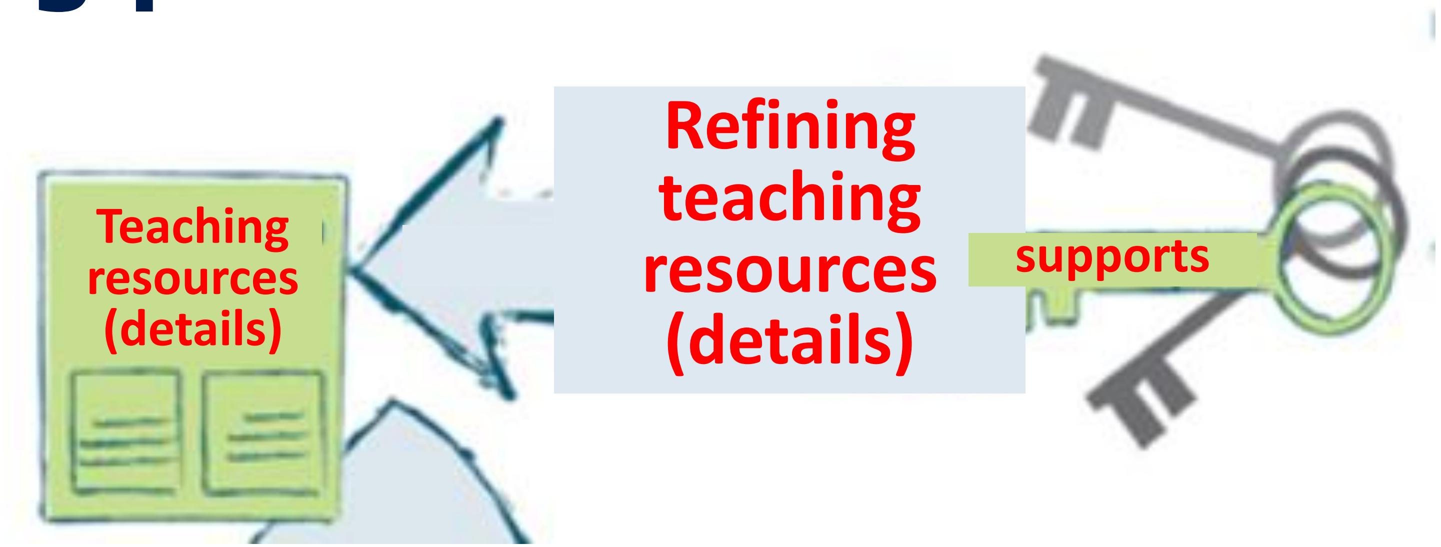
Brings further informations about the specification

(creation of a precise & concrete context)

Overview & corresponding pattern



⇒



Teaching resources (details)

Refining teaching resources (details)

supports

To be unambiguous and useful
(long term documentation)

→ **Focus** on business functionality

→ **precise**, self-explanatory

→ **testable**, in domain language

Teaching resources details

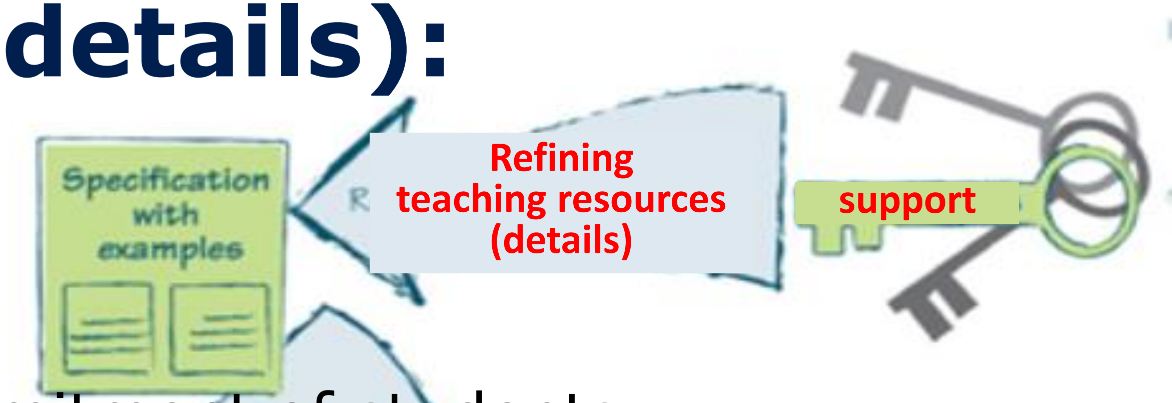
→ **Focus** on a specific competency & time-boxed, starter kit

→ **Precise :** choice of application domain to promote the commitment of students

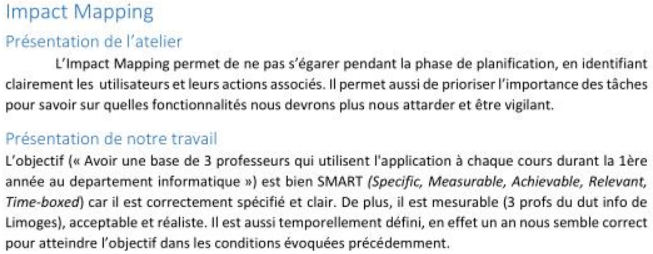
→ **Testable :** with a definition of an outcome

Outcome :
Describes a way to verifiy and validate that the required learning goals are well-achieved

Refining a teaching resource (details): Exemple


Specification with examples
Refining teaching resources (details)
support

→ Focus on a specific competency & time-boxed
→ Starter kit
→ Choice of application domain to promote the commitment of students
→ Various outcomes : oral feedback, photos, specific artifact, summary



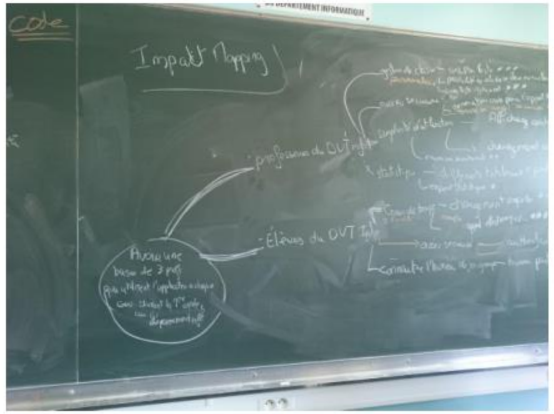
Impact Mapping

Présentation de l'atelier

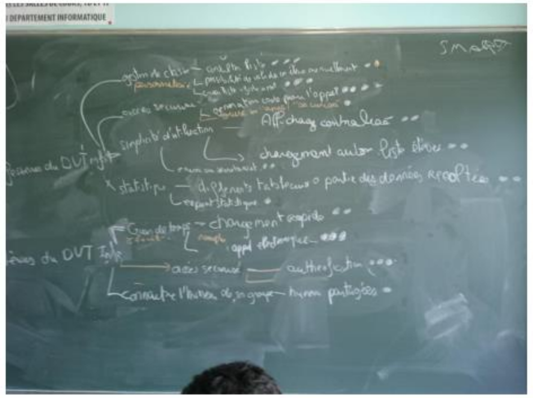
L'Impact Mapping permet de ne pas s'égarer pendant la phase de planification, en identifiant clairement les utilisateurs et leurs actions associés. Il permet aussi de prioriser l'importance des tâches pour savoir sur quelles fonctionnalités nous devrons plus nous attarder et être vigilant.

Présentation de notre travail

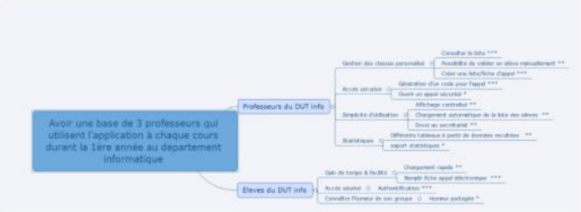
L'objectif (« Avoir une base de 3 professeurs qui utilisent l'application à chaque cours durant la 1ère année au departement informatique ») est bien SMART *(Specific, Measurable, Achievable, Relevant, Time-boxed*) car il est correctement spécifié et clair. De plus, il est mesurable (3 profs du dut info de Limoges), acceptable et réaliste. Il est aussi temporellement défini, en effet un an nous semble correct pour atteindre l'objectif dans les conditions évoquées précédemment.



Ci-dessus et sur la page suivante, notre Impact Mapping réalisé en cours.

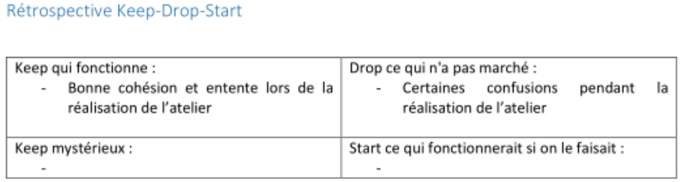


Ci –dessous, la version Xmind.


Avoir une base de 3 professeurs qui utilisent l'application à chaque cours durant la 1ère année au departement informatique
Professeurs du DUT info
Gestion des classes personnalisé
Consulter la liste ***
Possibilité de valider un élève manuellement **
Créer une liste/fiche d'appel ***
Accès sécurisé
Génération d'un code pour l'appel ***
Ouvrir un appel sécurisé *
Simplicité d'utilisation
Affichage centralisé **
Chargement automatique de la liste des élèves **
Envoi au secrétariat **
Statistiques
Différents tableaux à partir de données recoltées **
export statistiques *
Eleves du DUT info
Gain de temps & facilité
Chargement rapide **
Simple fiche appel électronique ***
Accès sécurisé
Authentification ***
Connaître l'humeur de son groupe
Humeur partagée *

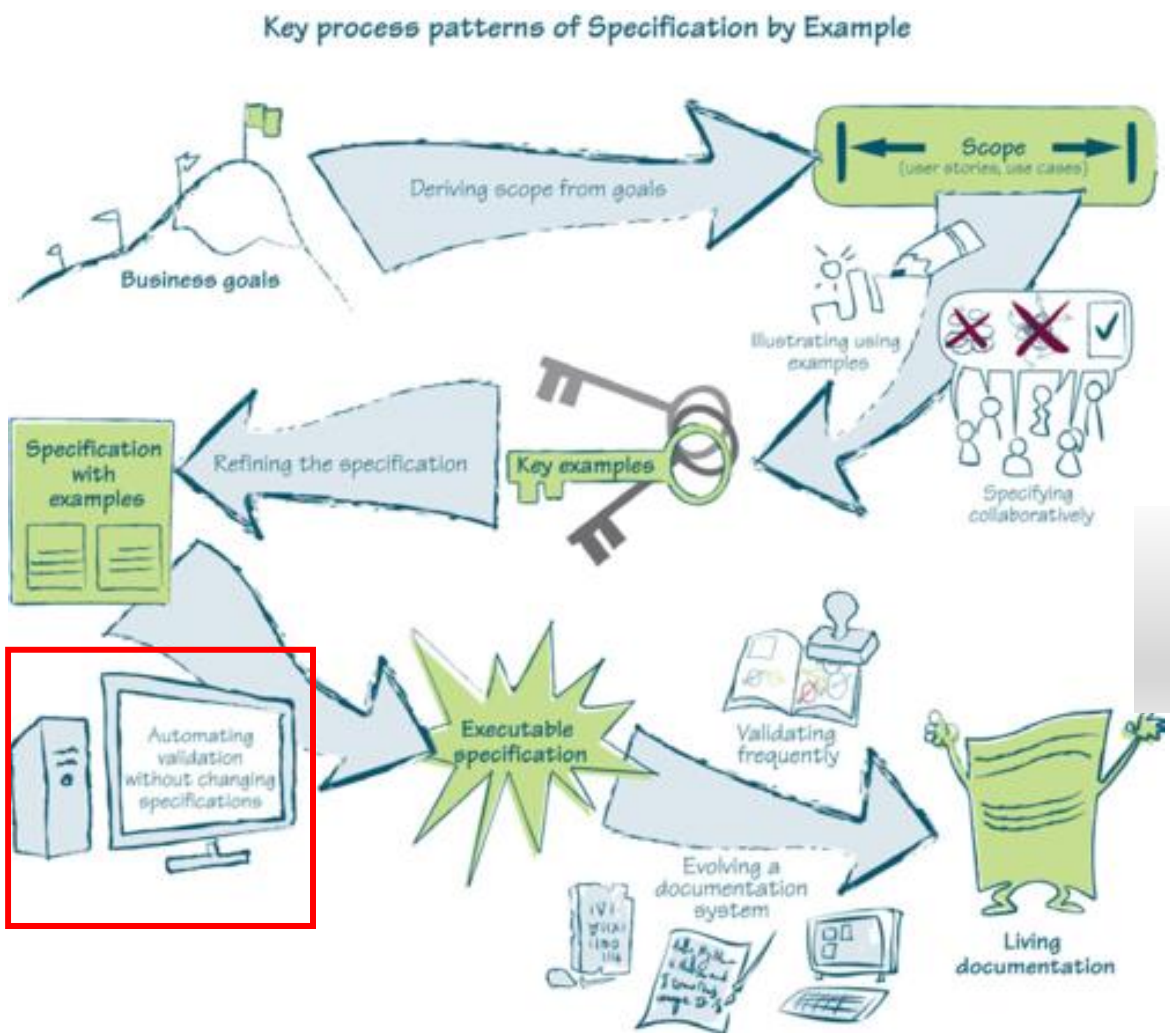
Commentaire sur l'atelier

Cet atelier nous a permis d'éclaircir les différentes fonctionnalités du projet, ainsi que la priorisation des différentes fonctionnalités, et les acteurs concernés.



Rétrospective Keep-Drop-Start

Keep qui fonctionne : - Bonne cohésion et entente lors de la réalisation de l'atelier	Drop ce qui n'a pas marché : - Certaines confusions pendant la réalisation de l'atelier
Keep mystérieux : -	Start ce qui fonctionnerait si on le faisait : -


Key process patterns of Specification by Example
Business goals
Deriving scope from goals
Scope
(user stories, use cases)
Illustrating using examples
Specifying collaboratively
Key examples
Refining the specification
Specification with examples
Automating validation without changing specifications
Executable specification
Validating frequently
Evolving a documentation system
Living documentation

Pattern 5 :

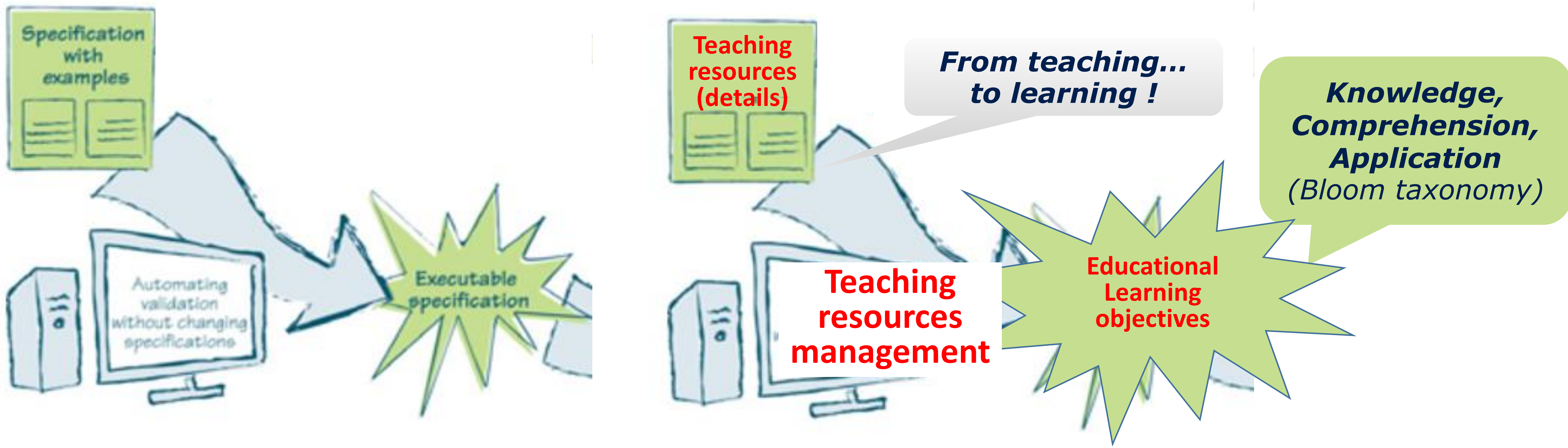
Automating validation without changing specification

Focus on automation,
as a solution of quick feedback

Specification with examples & automation testing tool
⇒ Executable specification

Overview & corresponding pattern

Thinking about automation,
Thinking about repetition of teaching resource
to aim educational learning objectives into differents levels)


Specification with examples
Automating validation without changing specifications
Executable specification

Teaching resources (details)

From teaching… to learning !

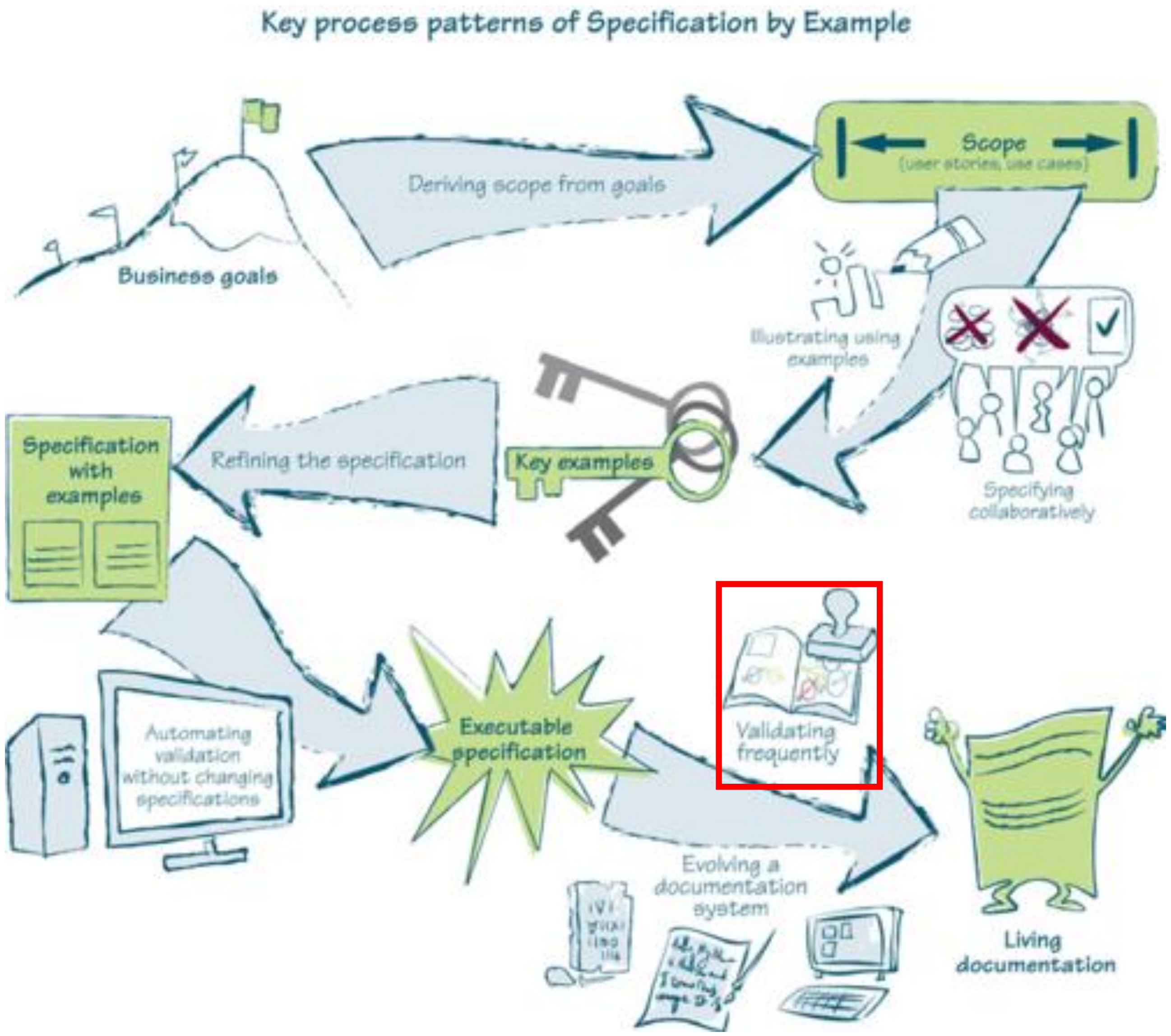
Knowledge, Comprehension, Application
(Bloom taxonomy)

Teaching resources management

Educational Learning objectives

Focus on resources management :
→ **schedule** : sequence of topics, sequence of various teaching resources for a specific topic
→ adaptation of schedule according to students behavior and feelings (feedback)

Example of repetition for a specific topic :
Lecture or game, collaborative workshop and application in a PBL

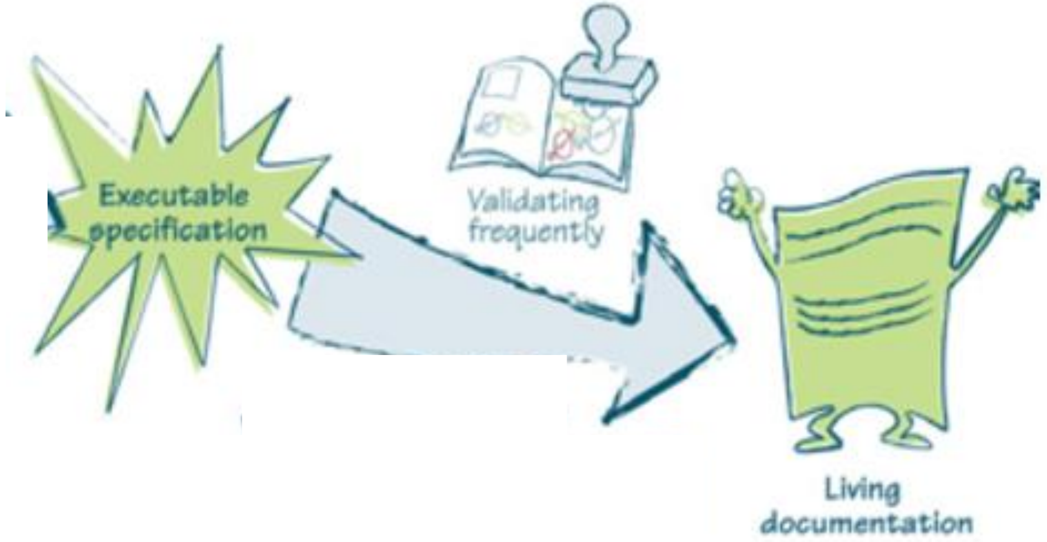

Key process patterns of Specification by Example
Business goals
Deriving scope from goals
Scope
(user stories, use cases)
Illustrating using examples
Specifying collaboratively
Key examples
Refining the specification
Specification with examples
Automating validation without changing specifications
Executable specification
Validating frequently
Evolving a documentation system
Living documentation

Pattern 6 :

Validating frequently

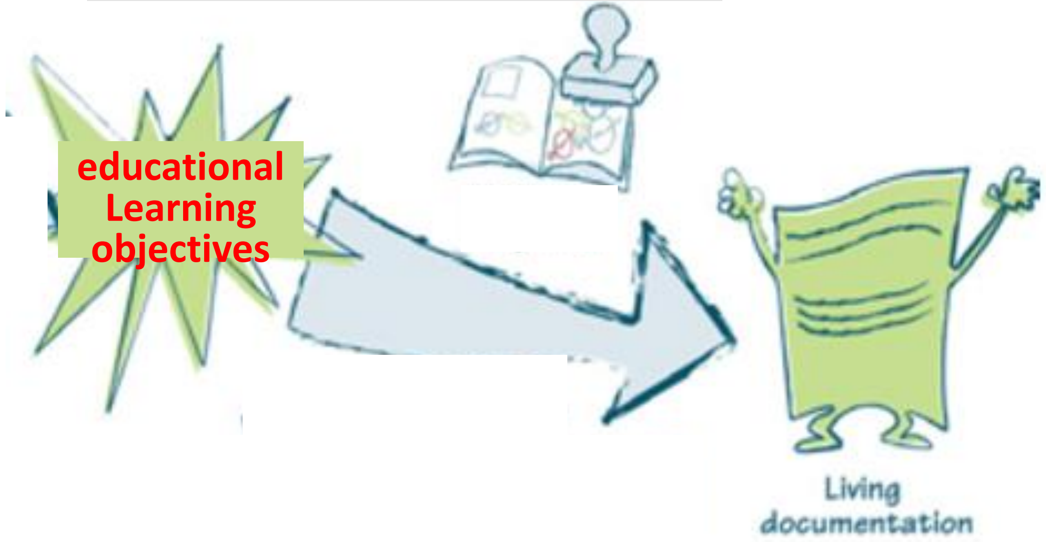
A continuous integration systems builds the product and runs the tests

Overview & corresponding pattern



⇒

Knowledge evaluation Process



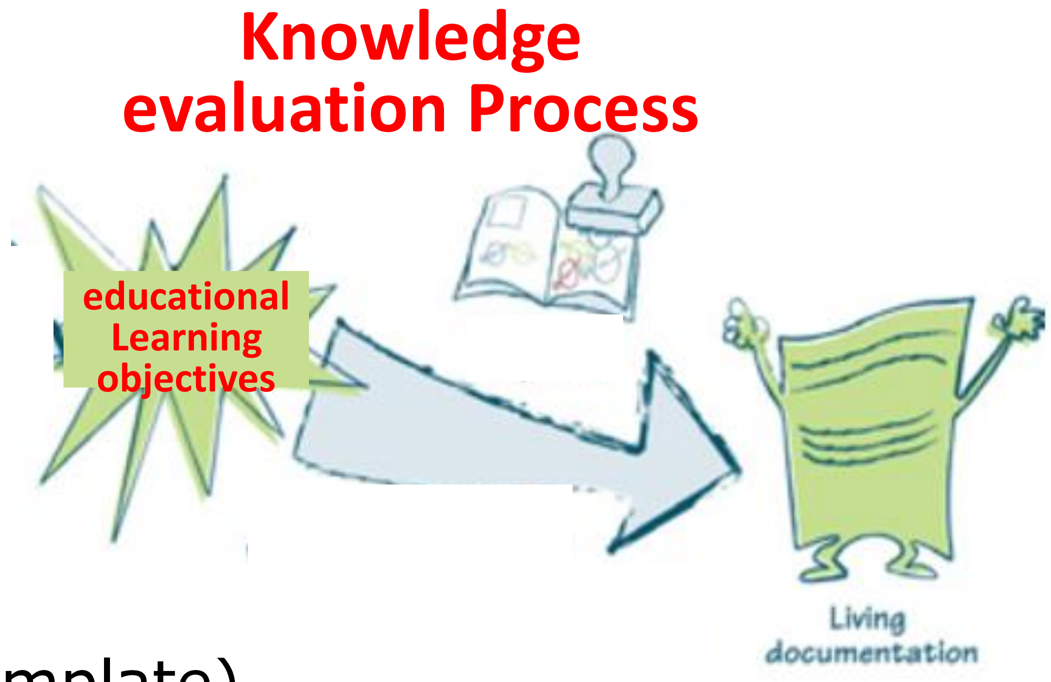
To ensure that the product stays right

→ validate executable specifications frequently
→ reducing unreliability
→ looking for ways to get faster feedbacks

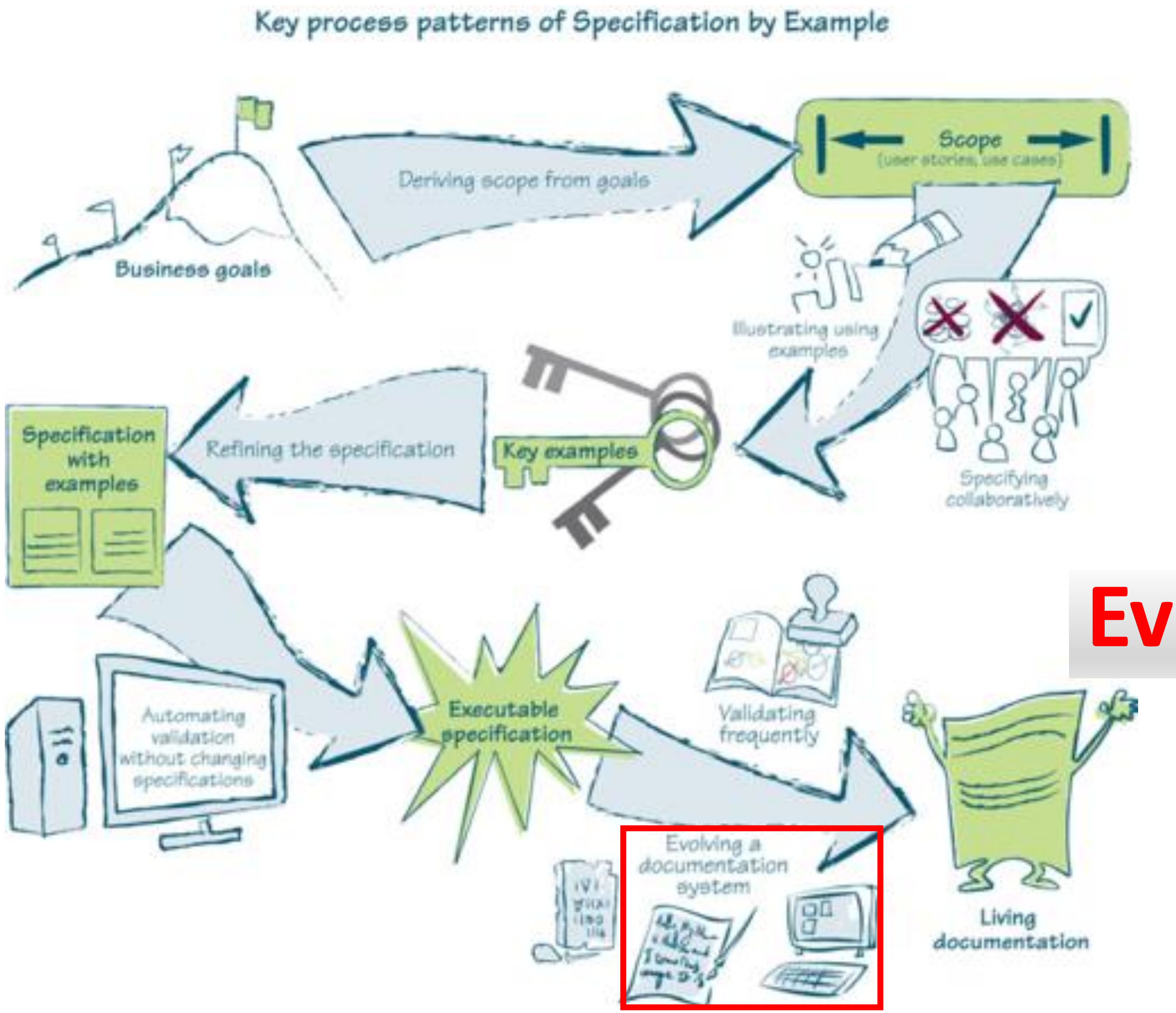
→ teacher : responsible for the reliability
→ students : responsible to get faster feedback

Practices : assignments, with or without grades, individual or collective, frequency

Knowledge evaluation Process Exemple



→ Evaluation at the end of each workshop
→ Summary of the workshop (based on a predefined template)
→ Report about PBL like a cookbook
→ collective grade in respect to CI process

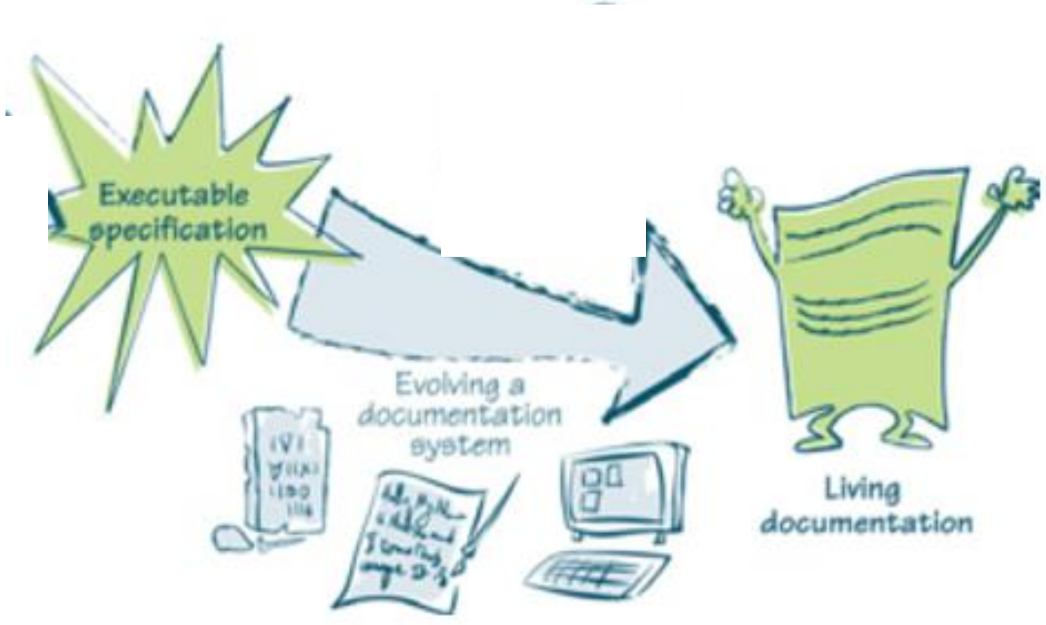

Key process patterns of Specification by Example
Business goals
Deriving scope from goals
Scope
(user stories, use cases)
Illustrating using examples
Specifying collaboratively
Key examples
Refining the specification
Specification with examples
Automating validation without changing specifications
Executable specification
Validating frequently
Evolving a documentation system
Living documentation

Pattern 7 :

Evolving a documentation system

Living documentation : artifact & the end-product

Overview & corresponding pattern



⇒

Living documentation

educational Learning objectives

Evolving a documentation academic

Living Documentation
(authoritative reliable source information on system)

→ Easy to access → repositories in a version control system

→ Easy to understand → by encouraging to share & update materials

→ Each change needs to be reflected → Be alive : notification system to alert all the skateholder when a new document is added or updated

Evolving a documentation academic Exemple

GitHub

Easily access on line public material course

IFTTT

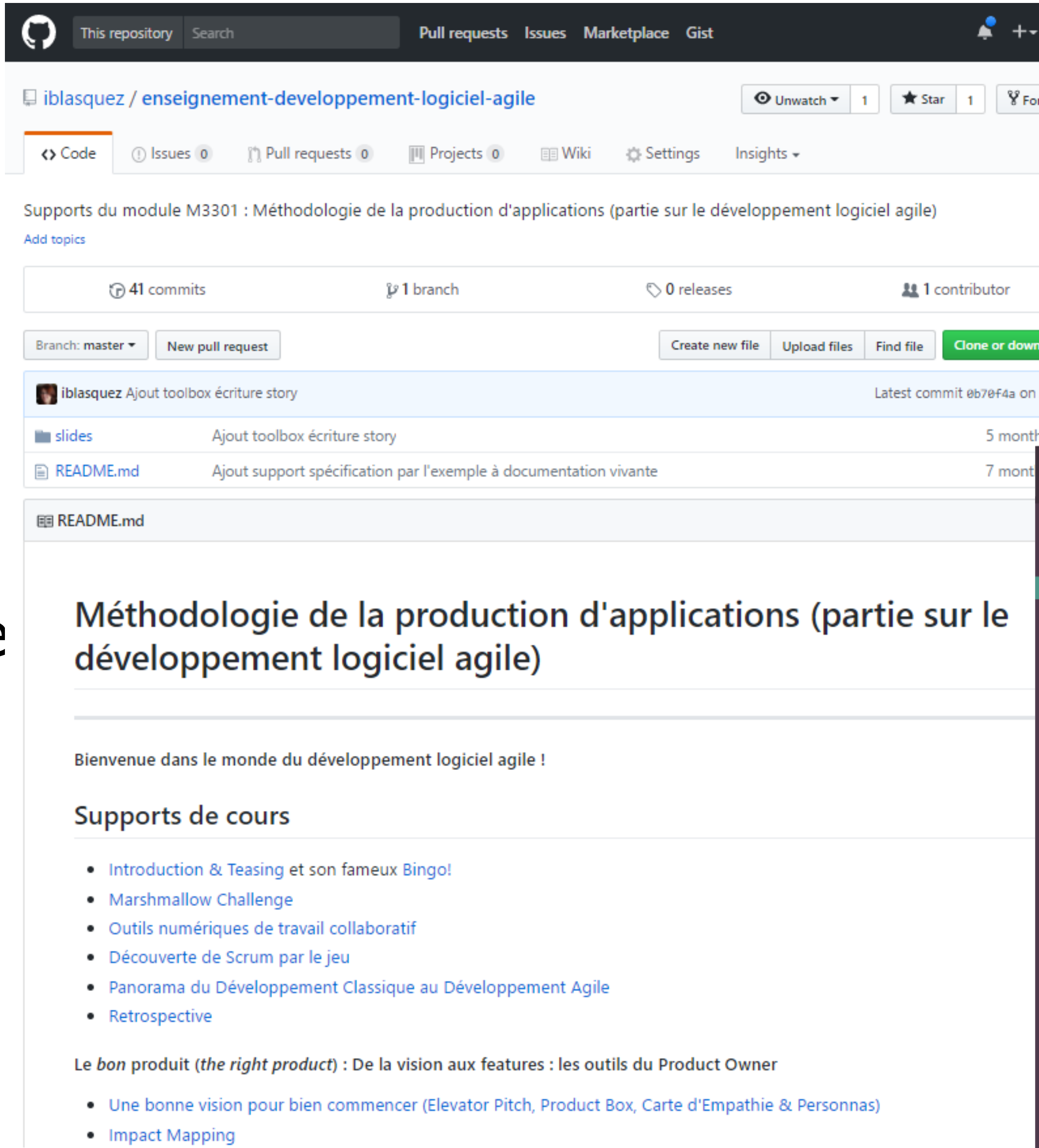
Recipe

if this then that

Trigger

Action

more understandable
& automatic notification system



This repository Search Pull requests Issues Marketplace Gist

iblasquez / enseignement-developpement-logiciel-agile

Unwatch 1 Star 1 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Settings Insights

Supports du module M3301 : Méthodologie de la production d'applications (partie sur le développement logiciel agile) Edit

Add topics

41 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

iblasquez Ajout toolbox écriture story Latest commit 0b70f4a on 25 Jan

slides	Ajout toolbox écriture story	5 months ago
README.md	Ajout support spécification par l'exemple à documentation vivante	7 mont

README.md

Méthodologie de la production d'applications (partie sur le développement logiciel agile)

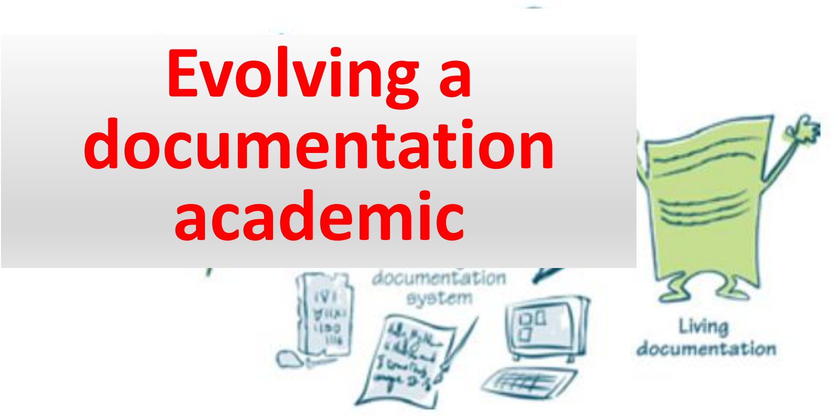
Bienvenue dans le monde du développement logiciel agile !

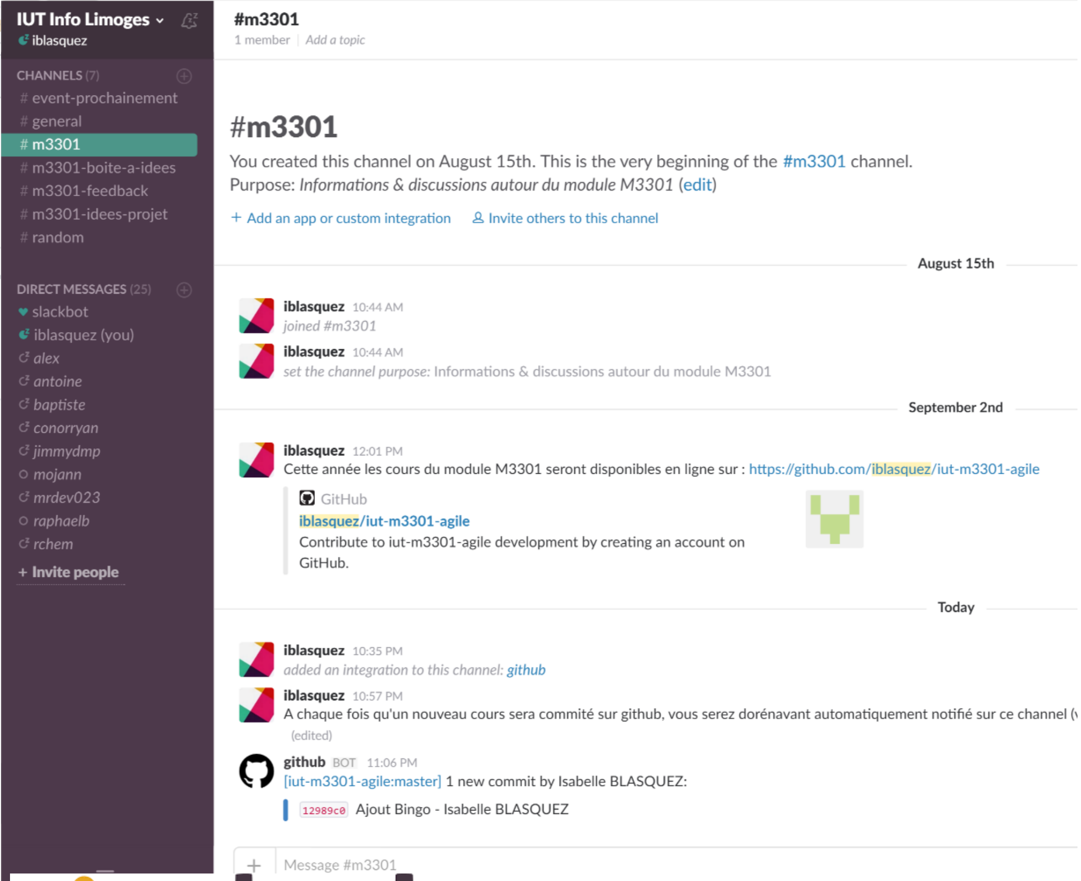
Supports de cours

- Introduction & Teasing et son fameux Bingo!
- Marshmallow Challenge
- Outils numériques de travail collaboratif
- Découverte de Scrum par le jeu
- Panorama du Développement Classique au Développement Agile
- Retrospective

Le *bon* produit (*the right product*) : De la vision aux features : les outils du Product Owner

- Une bonne vision pour bien commencer (Elevator Pitch, Product Box, Carte d'Empathie & Personnas)
- Impact Mapping





IUT Info Limoges
iblasquez

CHANNELS (7)
- # event-prochainement
- # general
- # m3301
- # m3301-boite-a-idees
- # m3301-feedback
- # m3301-idees-projet
- # random

DIRECT MESSAGES (25)
- slackbot
- iblasquez (you)
- alex
- antoine
- baptiste
- conorryan
- jimmydmp
- mojann
- mrdev023
- raphaelb
- rchem

+ Invite people

#m3301
1 member | Add a topic

#m3301

You created this channel on August 15th. This is the very beginning of the #m3301 channel.
Purpose: *Informations & discussions autour du module M3301* (edit)

+ Add an app or custom integration   Invite others to this channel

August 15th

iblasquez 10:44 AM
joined #m3301

iblasquez 10:44 AM
set the channel purpose: Informations & discussions autour du module M3301

September 2nd

iblasquez 12:01 PM
Cette année les cours du module M3301 seront disponibles en ligne sur : https://github.com/iblasquez/iut-m3301-agile

GitHub
iblasquez/iut-m3301-agile
Contribute to iut-m3301-agile development by creating an account on GitHub.

Today

iblasquez 10:35 PM
added an integration to this channel: github

iblasquez 10:57 PM
A chaque fois qu'un nouveau cours sera commité sur github, vous serez dorénavant automatiquement notifié sur ce channel (
(edited)

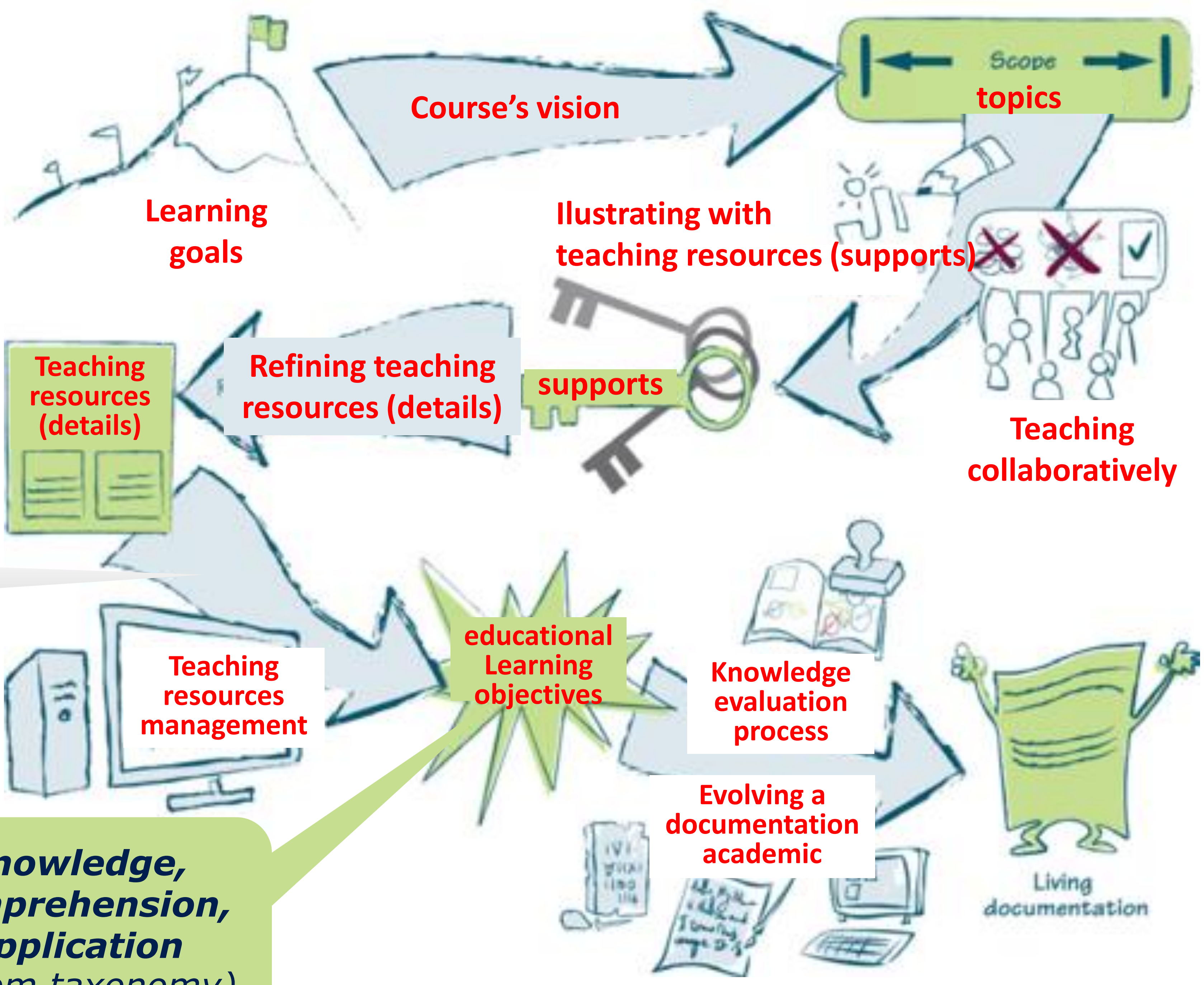
github BOT 11:06 PM
[iut-m3301-agile:master] 1 new commit by Isabelle BLASQUEZ:
12989c0 Ajout Bingo - Isabelle BLASQUEZ

Message #m3301

slack

for Education Purpose

Key process patterns of Specification by Example



Learning goals

Course’s vision

topics

Ilustrating with teaching resources (supports)

supports

Teaching collaboratively

Teaching resources (details)

Refining teaching resources (details)

Teaching resources management

educational Learning objectives

Knowledge evaluation process

Evolving a documentation academic

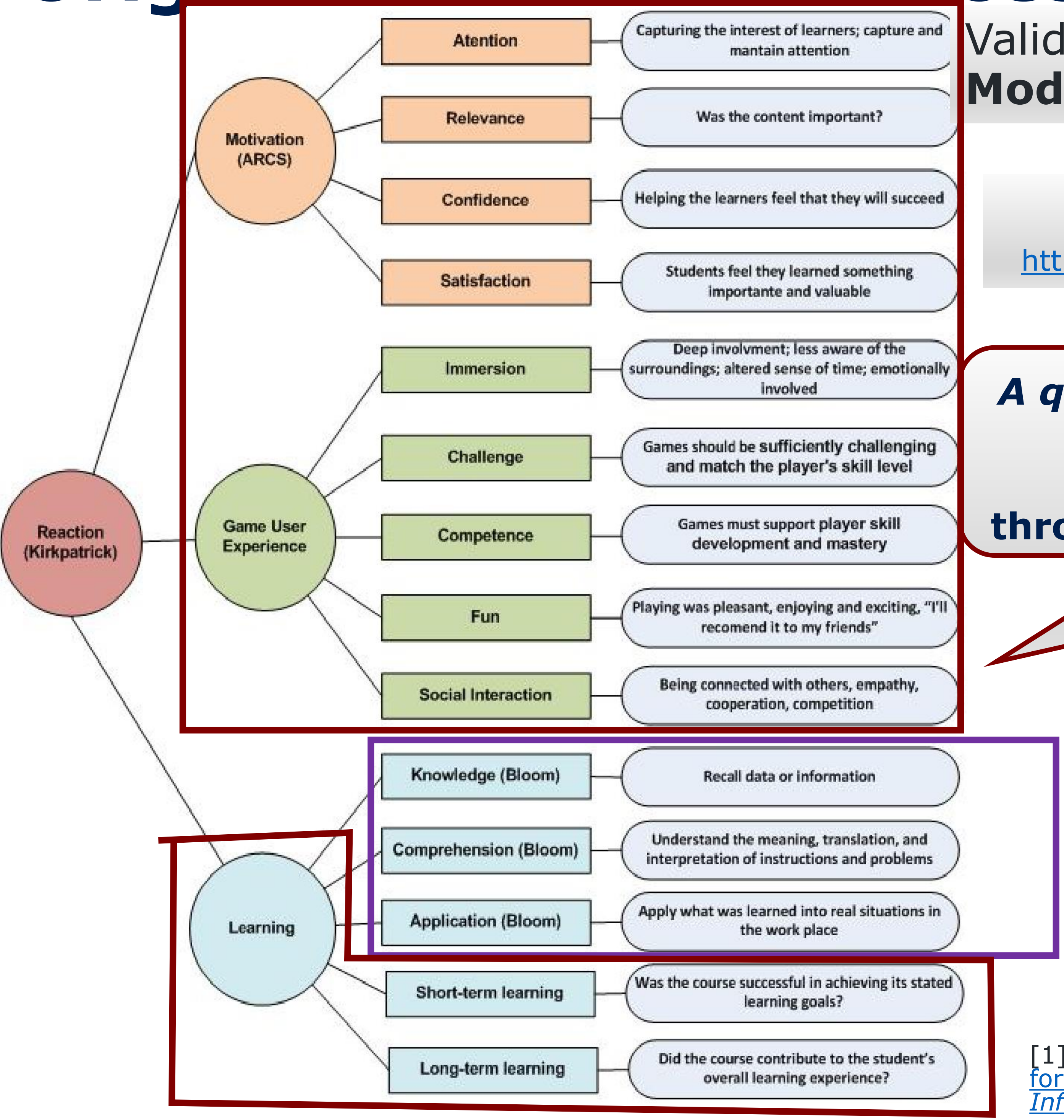
From teaching… to learning !

Knowledge, Comprehension, Application
(Bloom taxonomy)

Validation

… Towards a right course …

Original Validation Process



Reaction (Kirkpatrick)

Motivation (ARCS)
- Atention: Capturing the interest of learners; capture and mantain attention
- Relevance: Was the content important?
- Confidence: Helping the learners feel that they will succeed
- Satisfaction: Students feel they learned something importante and valuable

Game User Experience
- Immersion: Deep involvment; less aware of the surroundings; altered sense of time; emotionally involved
- Challenge: Games should be sufficiently challenging and match the player's skill level
- Competence: Games must support player skill development and mastery
- Fun: Playing was pleasant, enjoying and exciting, "I'll recomend it to my friends"
- Social Interaction: Being connected with others, empathy, cooperation, competition

Learning
- Knowledge (Bloom): Recall data or information
- Comprehension (Bloom): Understand the meaning, translation, and interpretation of instructions and problems
- Application (Bloom): Apply what was learned into real situations in the work place
- Short-term learning: Was the course successful in achieving its stated learning goals?
- Long-term learning: Did the course contribute to the student's overall learning experience?

Validation process adapted from an original model
Model for the Evaluation of Educational Games [1]

Evaluation kit of the model available on
http://www.gqs.ufsc.br/meega-a-model-for-evaluating-educational-games/

A questionnaire based on Kirkpatrick Evaluation
27 items asking motivation, user experience, and learning
through 11 dimensions (*attention, relevance, confidence,…*)

A questionnaire based on Bloom Taxonomy
Evolution of learning in the competencies taught
before and after the game

[1] C. G. von Wangenheim, R. Savi, and A. F. Borgatto. Deliver! - an educational game for teaching earned value management in computing courses. *Inf. Softw. Technol.*, 54(3):286–298, 2012

Adapted validation process to our case study (1/3)

1. Revise/adapt the objective of the evaluation

→ A definition of **quality** in the context of educational game in [1]
"a game is high-quality, if it provides a positive learning effect, motivates students to study and provides a pleasant and engaging learning experience."

→ Extension of this definition to the ***right* course** whose the expectations in terms of **learning**, **collaboration**, **commitment** and **happiness** to work are similar.

→ The basic hypothesis is that the right course contributes positively to achieving theses learning objectives.

[1] C. G. von Wangenheim, R. Savi, and A. F. Borgatto.
Scrumia : An educational game for teaching scrum in computing courses.
Journal of Systems and Software, 86(10):2675–2687, 2013

Adapted validation process to our case study (2/3)

2. Revise/adapt the research

→ **research strategy** is based on the ***quality*** of a ***right course***.

→ **Our case study** :

- ***Agile Software Project Management Course*** *presented as example*
- ***80 two-year French undergraduates*** *(technical college specialized in Computer Technology)*
- *10 weeks with 2 * 2-hours sessions per week*

→ **Overview of Bloom Taxonomy** (knowledge-comprehension-application)

- Doing Agile & Being Agile

→ **Overview of Kirkpatrick** :

- Terminology adapted from game to course
- Only 21 items to only focus a set of teaching resources

Adapted validation process to our case study (3/3)

3. Plan the evaluation

→ Course designed & delivered in the fall 2015

4. Execute the evaluation collecting data via questionnaire

→ on line via google form (http://unil.im/sondage3301 in french)

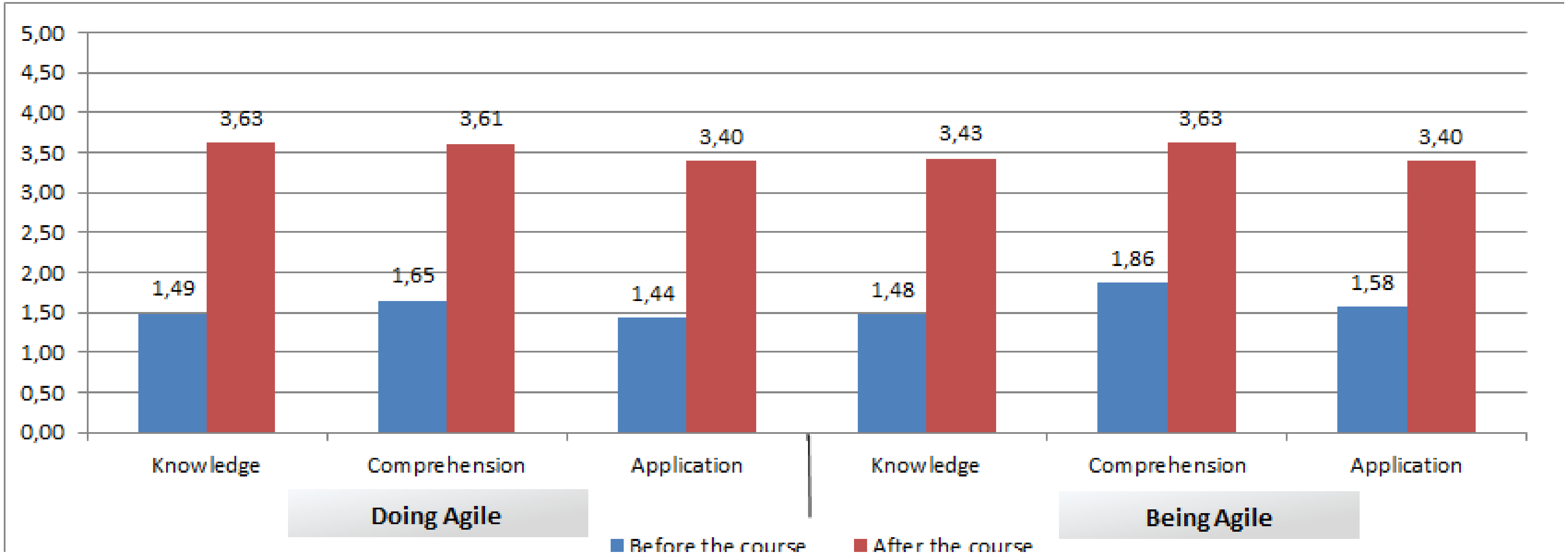
→ Each student anonymously filled out these questionnaires

→ once at the end of the course, **totalling 80 participants**.

5. Analyze the data collected using the analysis template

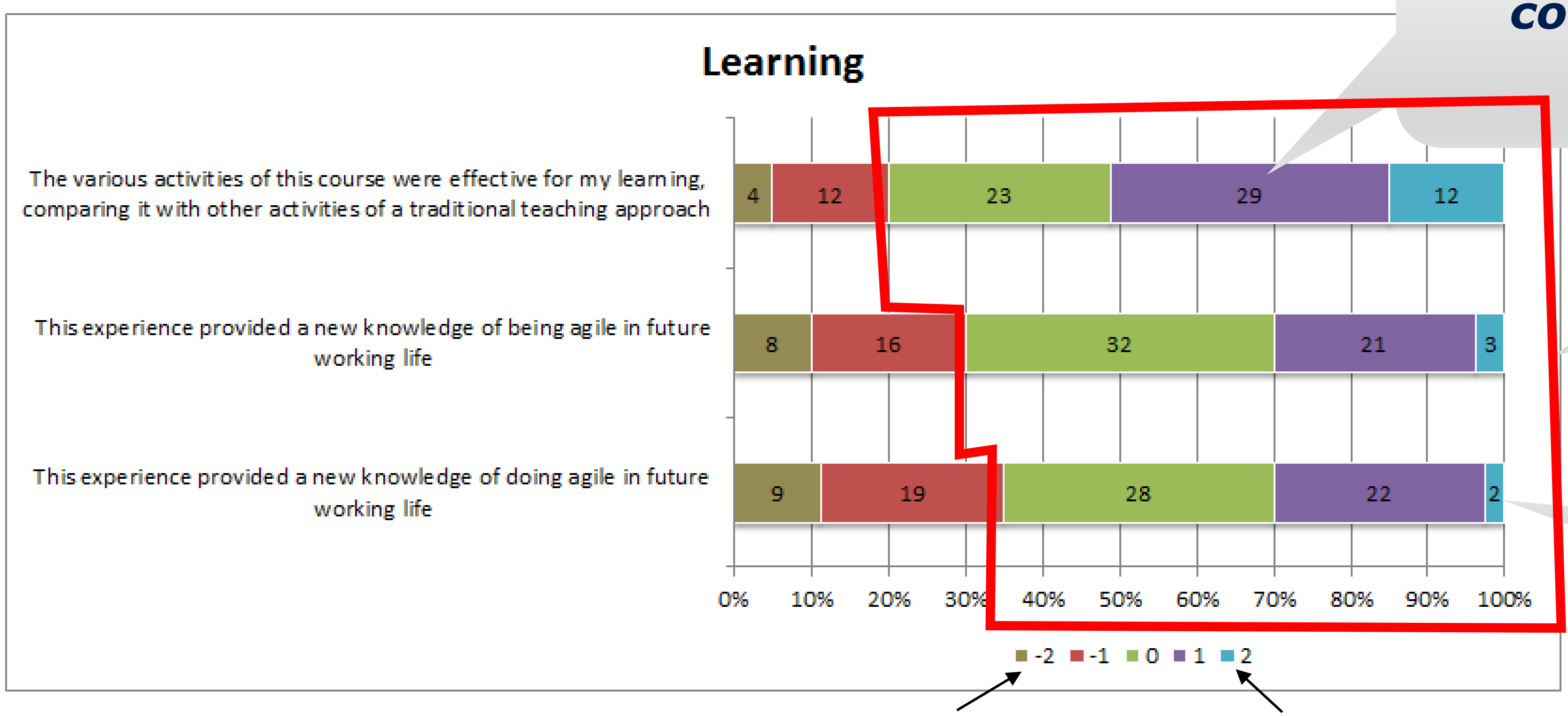
→ collected data analyzed through frequency diagrams in order to identify which are the most positive and negative aspects of the game with respect to each sub component/dimension.

Results : *Bloom Taxonomy*



Course has offered the opportunity to learn not only technical skills but also some values in a new mindset

Results *Kirkpatrick Evaluation*



Strongly disagree

Strongly agree

Short-term

Effectiveness of various teaching resources comparing it with traditional teaching approach activities

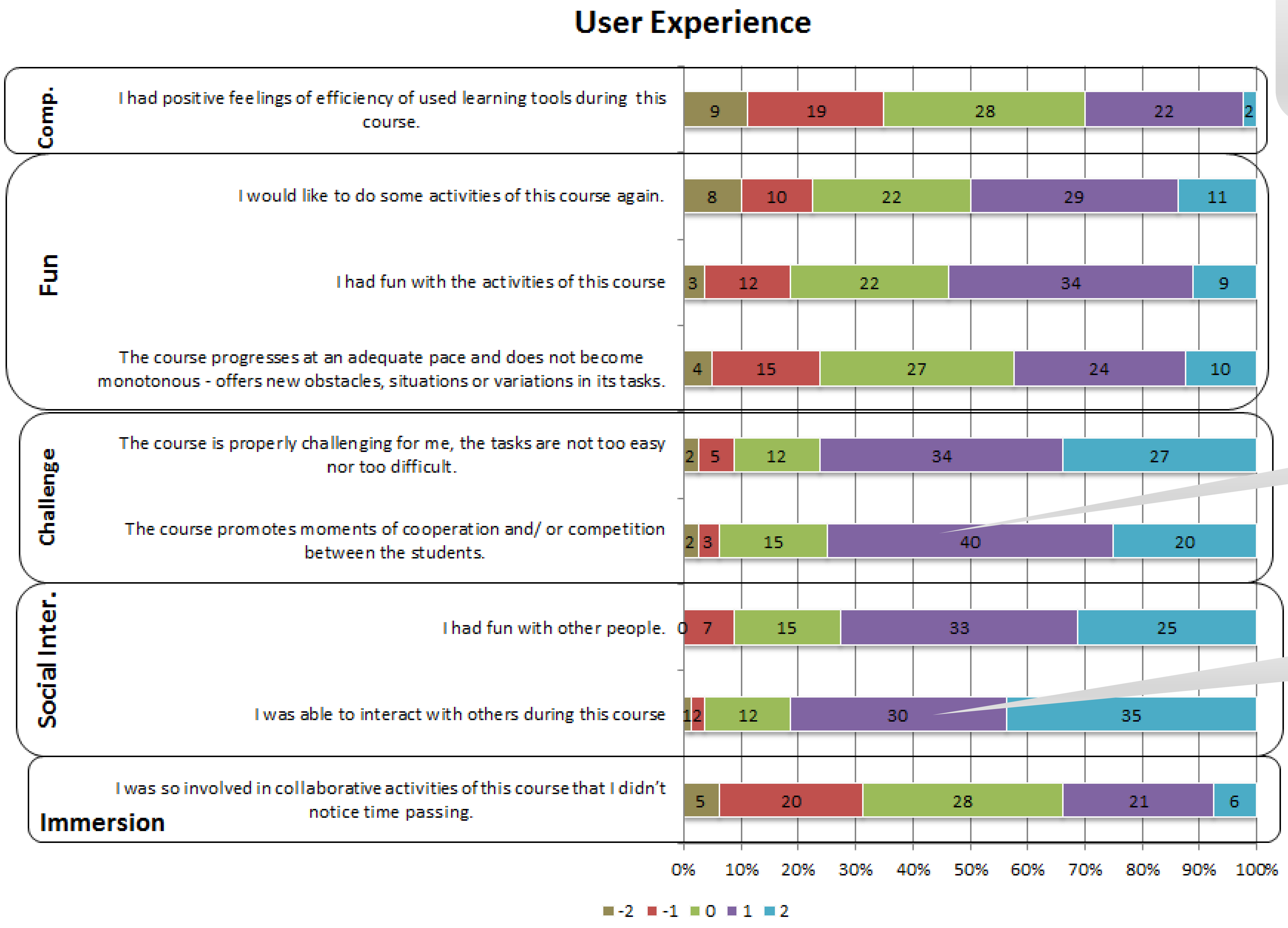
being agile in the future

doing agile in the future

Long-term

Course offers the opportunity to learn not only technical skills but also some values in a new mindset

Results *Kirkpatrick Evaluation*

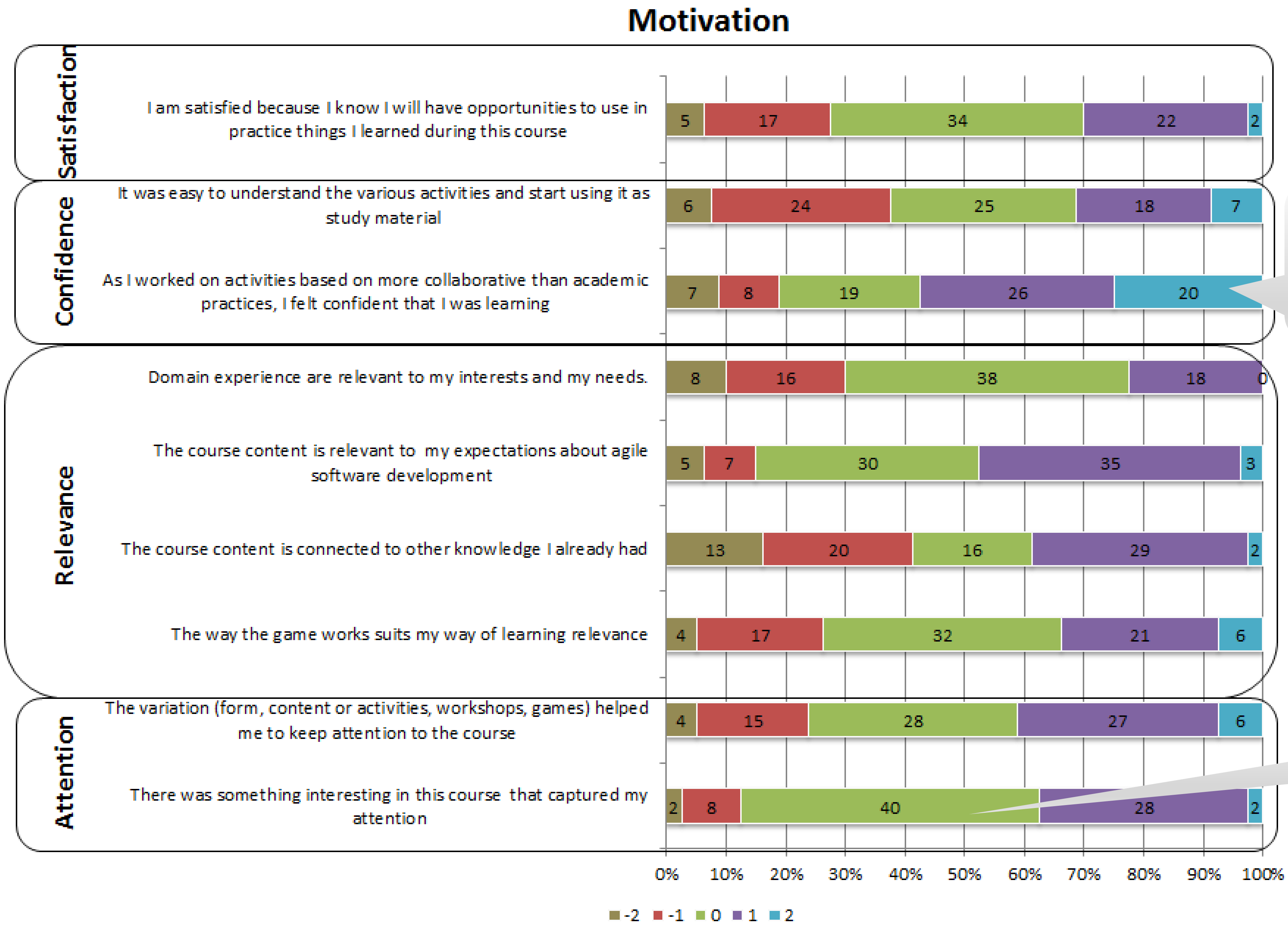


Positive results in terms of fun, challenge, social interaction

Moment of cooperation

Fun while interacting with others students

Results *Kirkpatrick Evaluation*



Feeling confident with more collaborative than academic practices

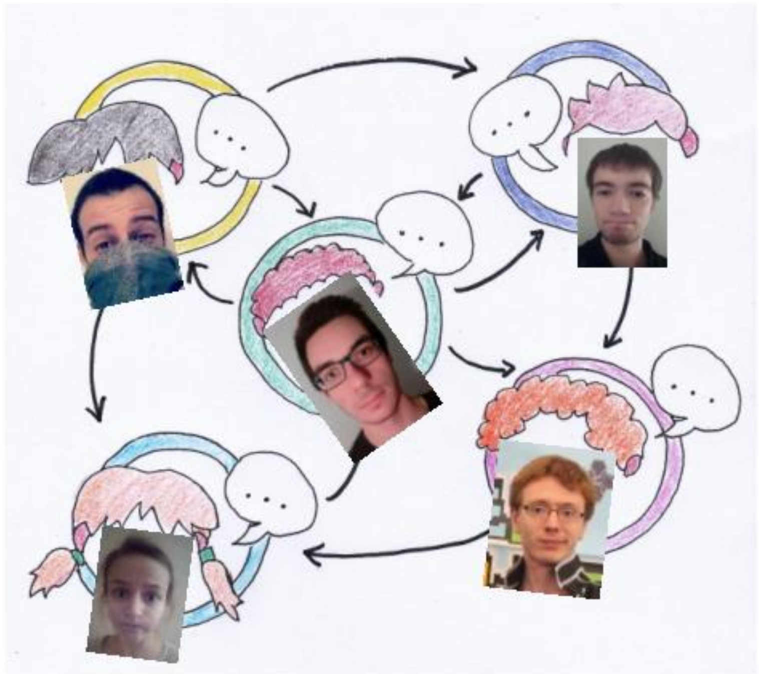
Better alignment : more attentive

A graphical conclusion from a student report …

CONCLUSION GRAPHIQUE

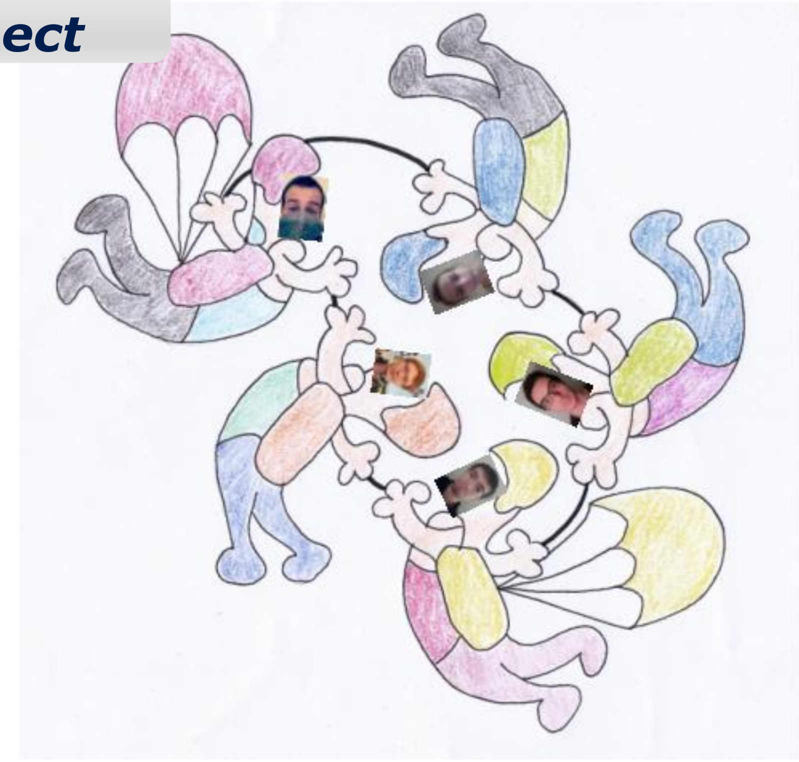
COMMUNIQUER

Communicate



Demonstrate respect

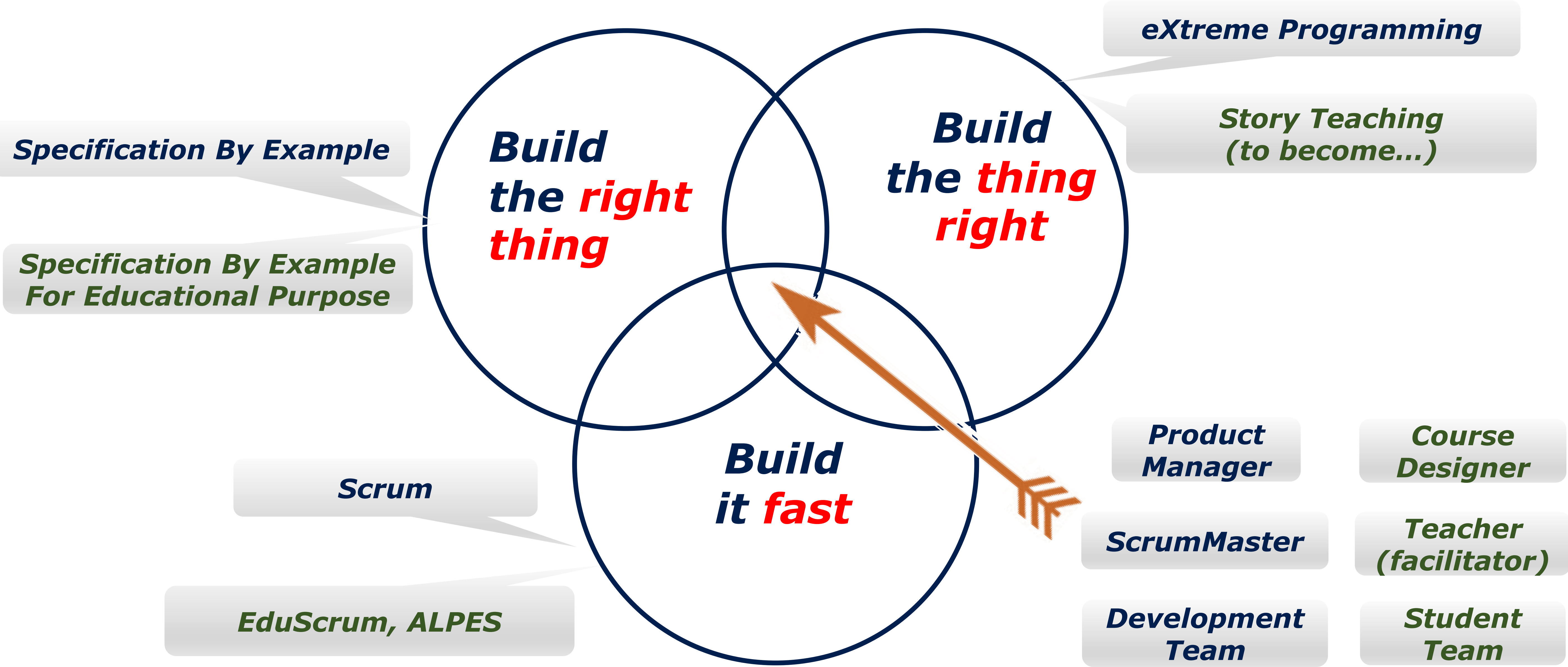
CONSIDERER L'AUTRE



Dare …

OSER SE LANCER

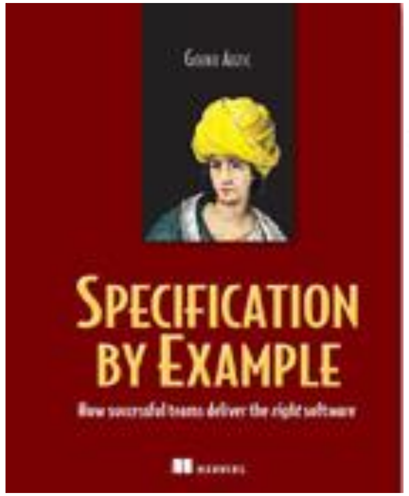
Teaching approaches in analogy to an agile software development & Future work



Three circles inspired by Henrick Kniberg

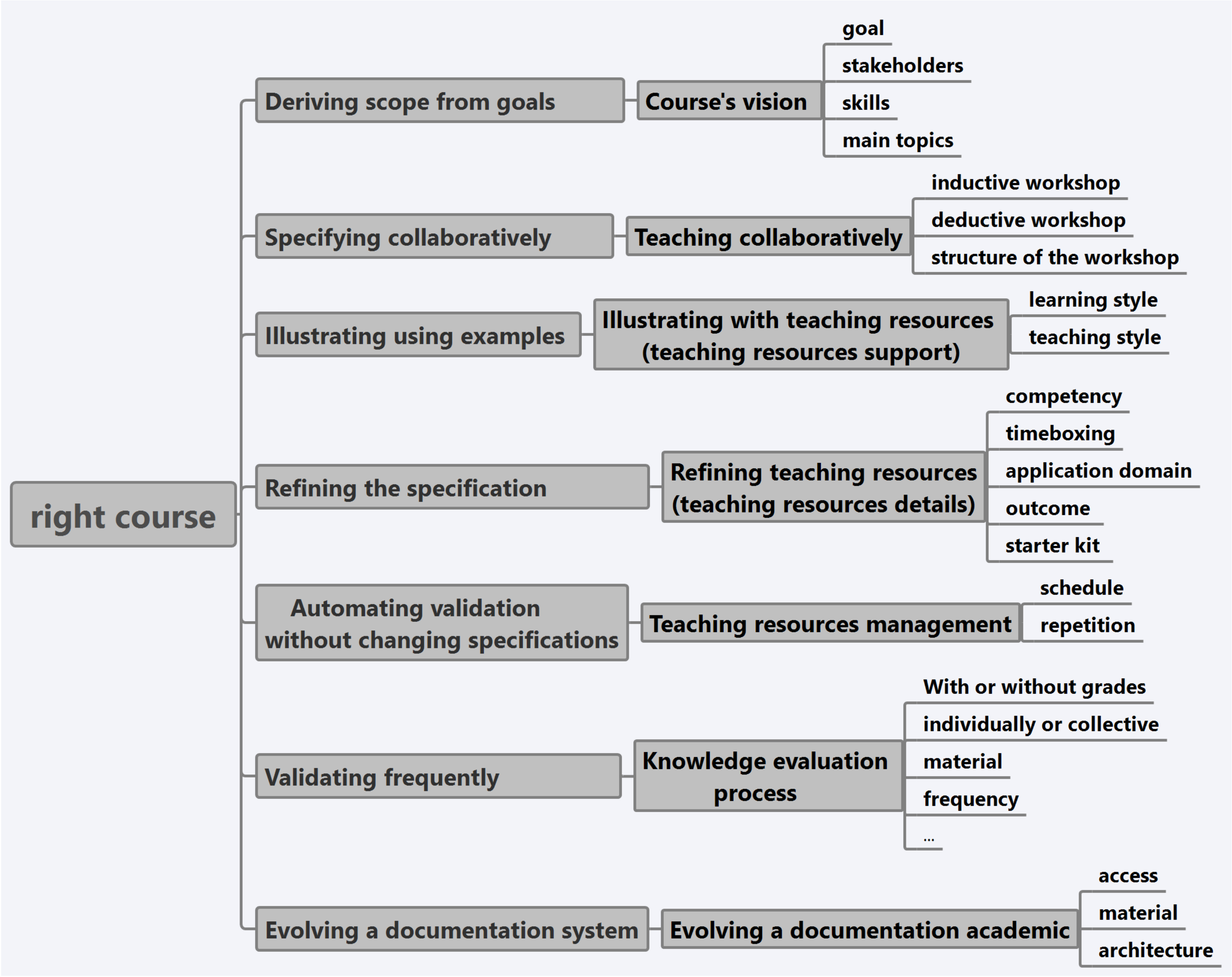
Thanks !

From 3 July 2013 to 3 July 2017



SPECIFICATION
BY EXAMPLE

Dear Isabelle
thank you very much
for your interest in
Spec by examples,
I hope you will be
able to use it to
improve your team!
Gojko Adzic
3. July 2013



right course

- Deriving scope from goals — Course's vision: goal, stakeholders, skills, main topics
- Specifying collaboratively — Teaching collaboratively: inductive workshop, deductive workshop, structure of the workshop
- Illustrating using examples — Illustrating with teaching resources (teaching resources support): learning style, teaching style
- Refining the specification — Refining teaching resources (teaching resources details): competency, timeboxing, application domain, outcome, starter kit
- Automating validation without changing specifications — Teaching resources management: schedule, repetition
- Validating frequently — Knowledge evaluation process: With or without grades, individually or collective, material, frequency, ...
- Evolving a documentation system — Evolving a documentation academic: access, material, architecture