



Agile Software Development Using Scrum

Viljan Mahnič

University of Ljubljana
Faculty of Computer and Information Science

viljan.mahnic@fri.uni-lj.si



Agenda

- Motivation
- Overview of Scrum
- User stories for requirements specification, release and Sprint planning
- Measuring progress
- Factors affecting success of Scrum projects
- Conclusions



Motivation

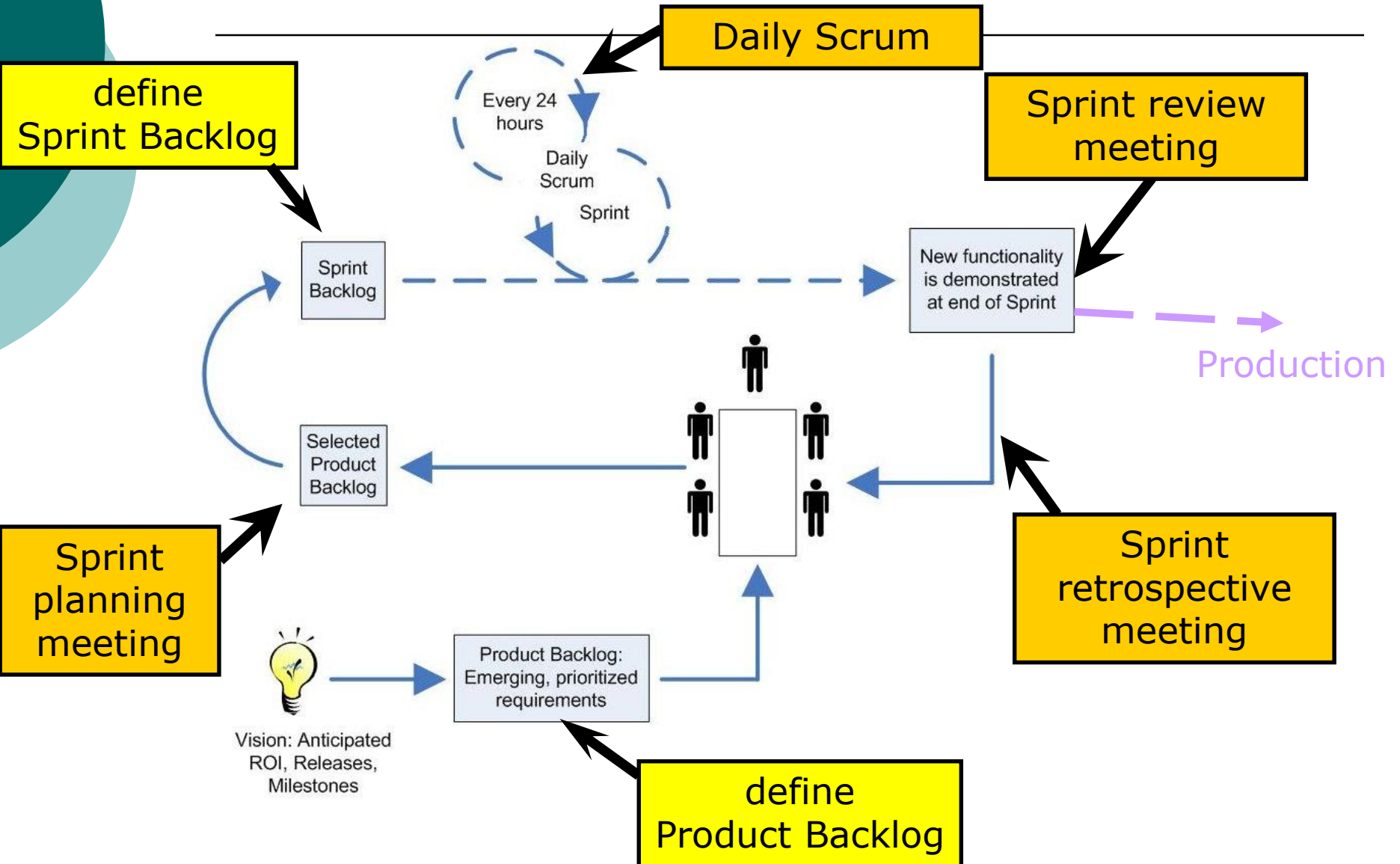
- Agile methods have been gaining wide acceptance
- Forrester, January 2010
 - 35% of respondents used an agile development process
- Gartner, December 2009
 - By 2012 agile development methods will be utilized in 80% of all software projects
- State of Agile Survey 2011
 - Scrum is the most widespread agile method (66%)
 - Loss of management control is one of the greatest concerns about adopting agile



Overview of Scrum

- Software development is too complex and unpredictable to be planned exactly in advance
 - Empirical process control must be applied to ensure visibility, inspection and adaptation
 - Iterative and incremental development process
- Scrum roles
 - Product Owner
 - Interests of everyone with a stake in the project
 - Team
 - Developing functionality
 - ScrumMaster
 - Managing the Scrum process

Overview of Scrum



User stories, release and Sprint planning

- A novel, lightweight approach to requirements specification
 - Detailed requirements need not be developed since these requirements inevitably change over time
- Each requirement is recorded as a user story consisting of three parts
 - Written description (used for planning and as a reminder)
 - Conversations about the story (to flesh out the details)
 - Acceptance tests (to determine when a story is “done”)
- Template

As a <type of user>, I want <some goal> so that <some benefit>.
- User stories shift the focus from writing about features to discussing them



User stories, release and Sprint planning

- Product Backlog
 - A set of all user stories currently known
 - For each story its priority and effort estimate must be defined
 - Planning poker: each story is assigned a number of story points
 - Initial velocity estimate: the number of story points that the Team can implement during a Sprint
- Release Plan
 - Created by allocating stories to Sprints strictly considering their priority and Team's estimated velocity
 - Provides an estimate of the approximate duration of a project and a rough content of each Sprint
- Sprint Backlog
 - Stories that have to be developed in the next Sprint
 - Each story is further decomposed into constituent tasks
 - Effort estimate (in hours) for each task is defined



Measuring progress

○ Derived measures

- Actual velocity
- Amount of work remaining till the end of project/release
- Amount of work remaining till the end of Sprint
- Earned value indexes (Mahnica and Vrana, 2007)
 - Schedule Performance Index
 - Cost Performance Index

○ Base measures

- Size of each user story in the Product Backlog (in story points)
- Amount of work spent on each task in the Sprint Backlog (in hours)
- Amount of work remaining on each task in the Sprint Backlog (in hours)

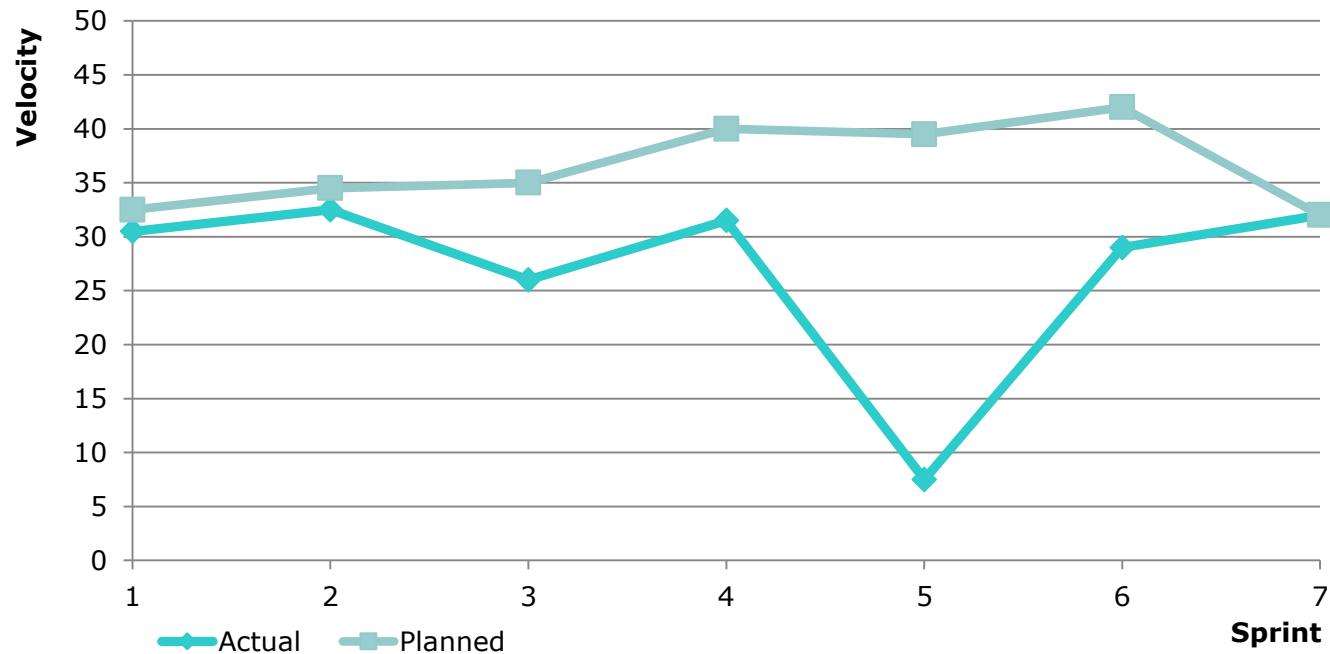


Measuring progress

- The largest Slovenian publishing company
 - Renew the web edition of their major daily newspaper
 - Introduce Scrum as the development process to their web applications department
- Basic characteristic
 - 7 months (May – November 2011), 9 Sprints, 6-8 people
 - Sprints lasted three weeks
 - Sprint Planning Meeting on Thursday
 - Sprint Review and Sprint Retrospective meetings on Tuesday of the third week of the Sprint
 - Daily Scrum Meetings took place every day
- Scrum was considered appropriate
 - Vague and changing requirements
 - Project's nature
 - Short time to deliver

Velocity

- Amount of work accomplished in each Sprint expressed in story points





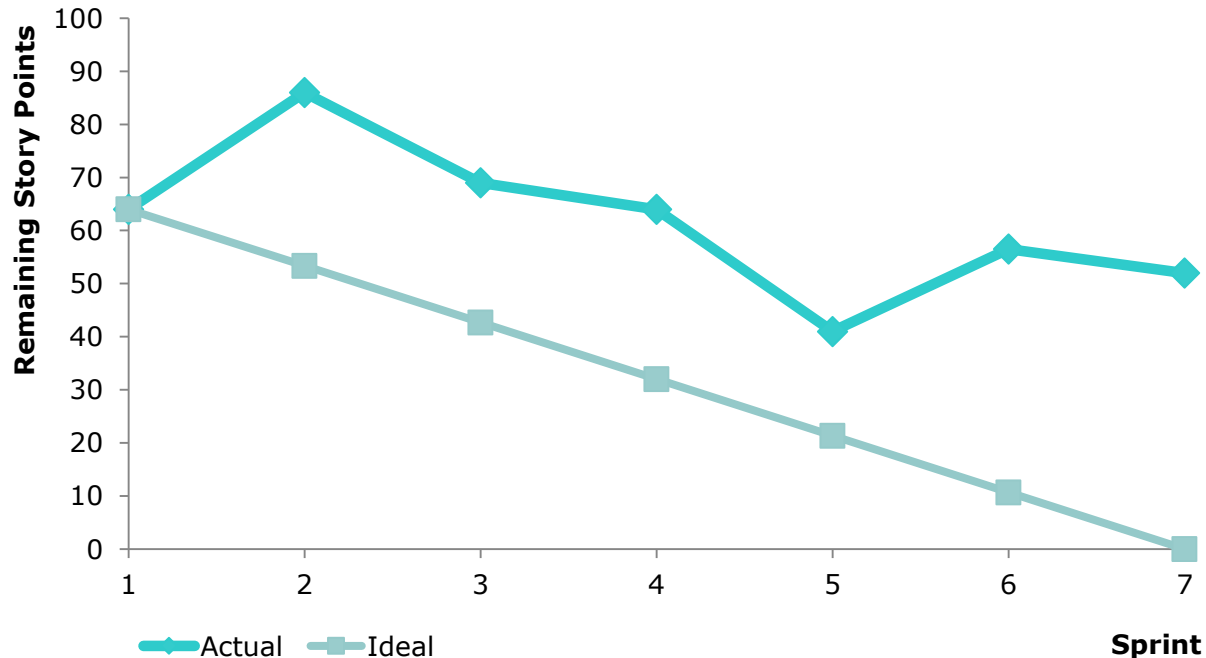
Velocity

- Analysis of velocity revealed two mistakes that should be avoided in Scrum projects
 - Planned velocity should be estimated considering the actual velocity of previous Sprints
 - There should be no changes in the development team in the middle of the project

Amount of work remaining till the end of the project/release

○ Release burndown chart

- The sum of story points of all unfinished stories at the beginning of each Sprint
- The trend line for work remaining indicates the most probable completion of work at a given point of time





Amount of work remaining till the end of the project/release

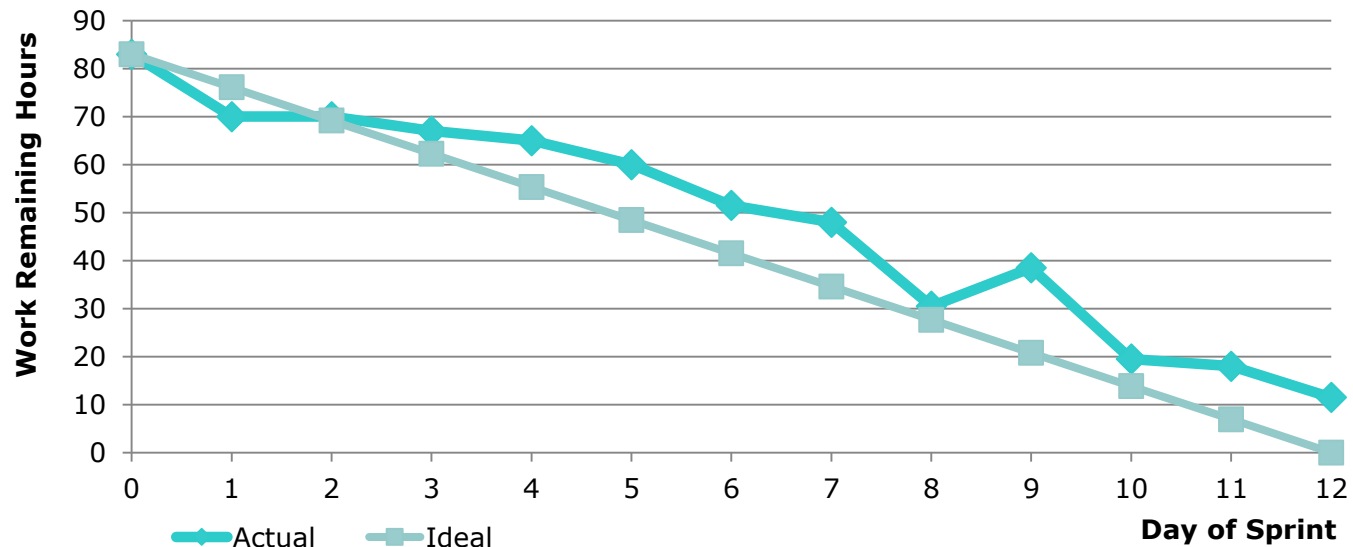
○ Release burndown chart

- The Team was not able to reduce the amount of work remaining quickly enough to complete the project in seven Sprints
- The main reason: emerging requirements that were constantly added by the Product Owner during the project
- Can be used to simulate the impact of removing functionality to get a more acceptable completion date
- Using this approach the company launched a reduced release after 9 Sprints.

Amount of work remaining till the end of the Sprint

○ Sprint burndown chart

- The horizontal axis shows the days of a Sprint, while the vertical axis shows the number of remaining working hours
- The chart is updated every day by aggregating the estimates of work remaining for all tasks in the Sprint Backlog
- The trend line of remaining working hours indicates whether the Team will accomplish the tasks by the end of the Sprint

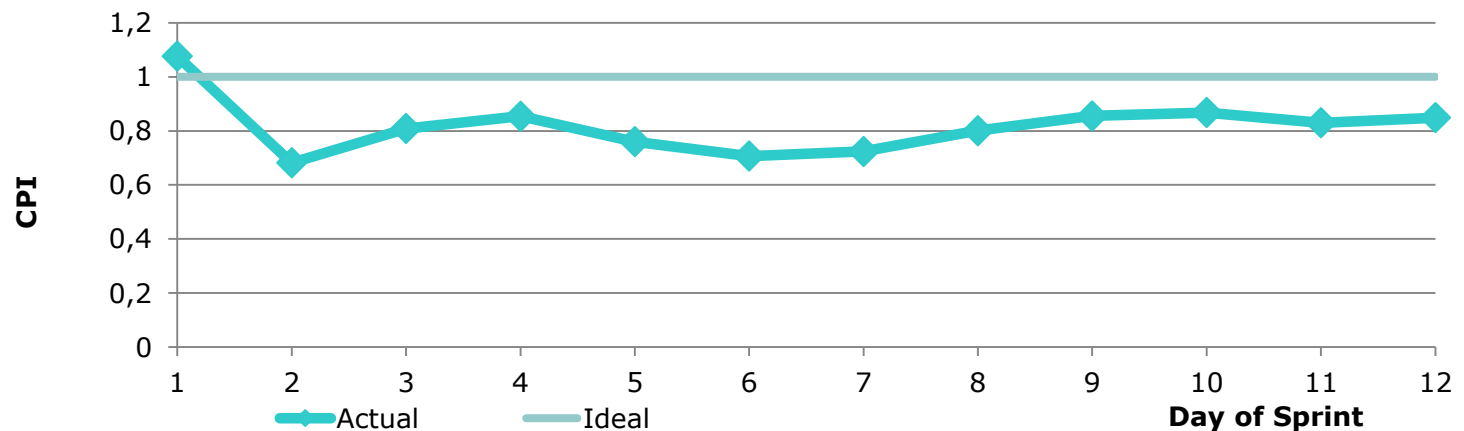
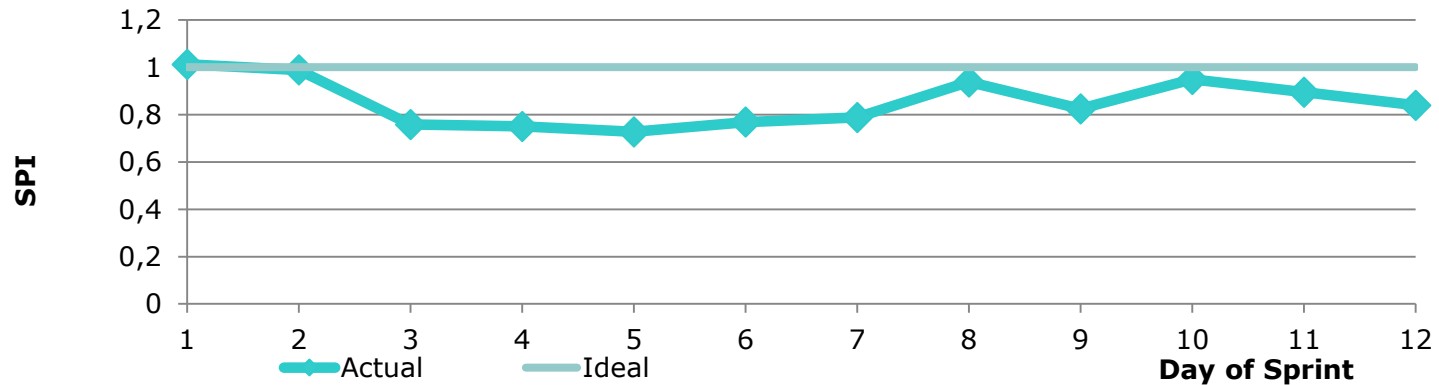




Computation of schedule and cost performance indexes

- Our approach provides the values of schedule performance and cost performance indexes on a daily basis
 - Immediate response in the case of deviation from the plan
 - Useful when longer Sprints are used
- Schedule performance index provides similar information as Sprint burndown chart
 - The value of SPI less than 1 indicates that the project is late
- Cost performance index provides information that is not available from any other measure
 - The value of CPI less than 1 indicates that the project exceeds planned costs
 - Computation of CPI requires collection of only one additional base measure, i.e., the number of hours spent on each task between two consecutive Daily Scrum meetings

Computation of schedule and cost performance indexes





Most important success factors

- A survey among Scrum users in Slovenia and abroad
- The most important factors
 - Team-work and communication among Team members
 - Good communication with the Product Owner
- Findings are in line with agile principles
 - The most efficient and effective method of conveying information is face-to-face conversation
 - Business people and developers must work together daily through the project
- Product Owner
 - Communicates the vision of what is to be developed
 - Defines the criteria by which the results will be judged
 - Provides details about user stories
 - Must be knowledgeable enough about Scrum to write, maintain and prioritize user stories



Most important success factors

- Common definition of “done”
 - Contributes significantly to the quality of the product
 - Ensures that each story is fully developed, tested, integrated, and resistant to user errors
 - A story can only be accepted by the Product Owner if it satisfies the definition of “done”
 - The definition should be established at the very beginning and followed strictly throughout the project
- Daily Scrum meetings
 - A means of empirical process control in order to ensure visibility, inspection and adaptation
 - Enable immediate detection and resolution of possible impediments
 - Must not be for reporting to the ScrumMaster, but for the Team members to inform each other about the current state of the project



Conclusions

- Our experience
 - Development of University of Ljubljana information system
 - Students' capstone projects
- Scrum is beneficial for all parties involved
- Team members
 - Scrum improves communication, and increases motivation and responsibility for the success of a project
- Product Owner and ScrumMaster
 - Software development process becomes visible, controllable, and manageable
 - All impediments are detected immediately and can be removed as soon as they emerge
- Customers
 - Scrum provides regular delivery of new functionality and frequent feedback on how the product actually works

Conclusions

○ Recent publications

- Mahnic, V., 2010. Teaching Scrum through team-project work: students' perceptions and teacher's observations. *International Journal of Engineering Education*, 26(1), 96–110.
- Mahnic, V., 2012a. A Capstone Course on Agile Software Development Using Scrum. *IEEE Transactions on Education*, 55(1), 99–106.
- Mahnic, V., 2012b. Introducing Scrum into the development of a news portal. Proceedings of 12th WSEAS International Conference on Applied Informatics and Communications (AIC'12), Istanbul, Turkey, 21st-23th August, 2012, pp. 109–114.
- Mahnic, V., Hovelja, T., 2012c. On using planning poker for estimating user stories. *Journal of Systems and Software*, 85(9), 2086–2095.



Thank you!

Questions ?