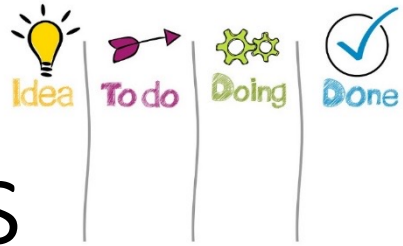


VO Softwareentwicklungsprozess

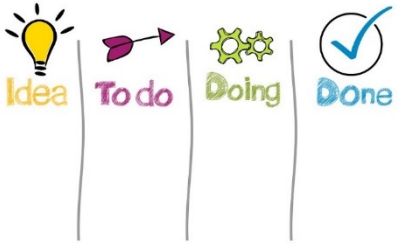
https://youtu.be/yIn-FaZMxC8

Alexander Felfernig und Trang Tran

Institut für Softwaretechnologie, Inffeldgasse 16b/2

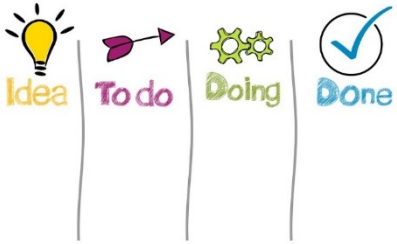
- Effort Estimation -

Effort Estimation: Motivation



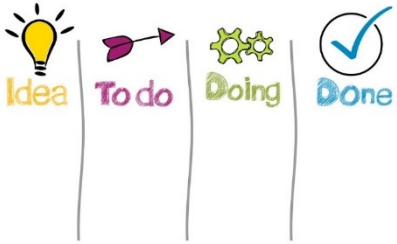
- Which requirements are feasible with the available resources?
- What will be the effort to complete a set of user stories?
- What will be the effort to complete the whole project?
- Knowing the effort is crucial for a business & clients!
- However: keep in mind that an estimate is an estimate and thus no reliable prediction!

Effort Estimation: Overview



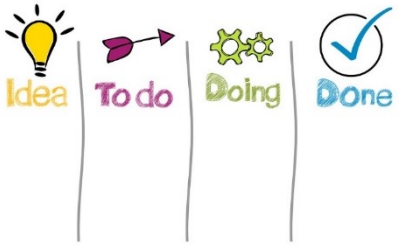
- **Analogy costing**: on the project or component level (availability of very similar projects needed)
- **Expert judgement**: a group of experts discuss the efforts and have to develop consensus regarding the estimate
- **Top down estimation**: based on basic parameters such as number and complexity of requirements (early stage)
- **Bottom-up estimation**: based on individual requirements
- **Our focus**: bottom-up + expert judgement

Requirement Size



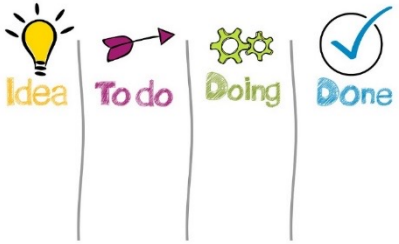
- Typical question of project managers: how long will it take to implement a requirement?
- A developer might answer: 2 days (this information could also be forwarded to customers!)
- However: not every detail is specified in a requirement and developers also spend time for emails, meetings, etc.
- Agile development: compare the size of individual requirements („story size“) instead of providing an absolute estimate!

Requirement Size vs. Duration



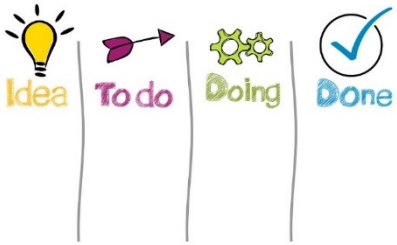
- Requirement A: 2 person days
- Requirement B: 3 person days
- High probability that effort related to requirement B is higher compared to the effort of requirement A
- However: no guarantee that A is ready in exactly 2 (person) days and B is ready in exactly 3 (person) days!
- Size of a requirement: „A has a lower implementation effort compared to B“ is a more reliable prediction.

Classification of Requirements



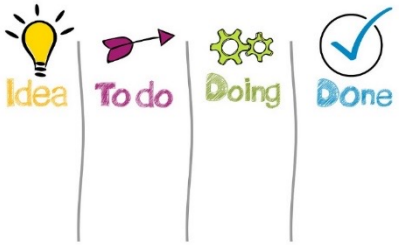
- Requirements sizes: e.g., small (S), average (M), large (L)
- Choose an average size requirement R (class M): all other requirements can be compared with R then …
- For example, if R is „registration“ (class M), then U „sign in“ is smaller (class S) and V „user profile definition“ is larger (class L)
- Thus, early estimation can be regarded as a classification task
- In real-world software projects: more than three classes!

Example Classification: Story Points



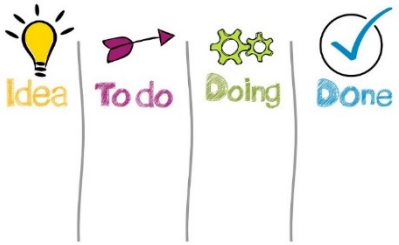
- Efforts of requirements/user stories expressed as story points
- If the effort (size) of R1 has been estimated with 2, its size is around ½ of a requirement R2 with an estimated size of 4
- Up to now: no statement about absolute duration, but relative duration (the development of R2 takes twice the time of R1)
- Especially in agile development (e.g., SCRUM): strict separation of requirement size and implementation duration estimation!

Story Points: Motivation



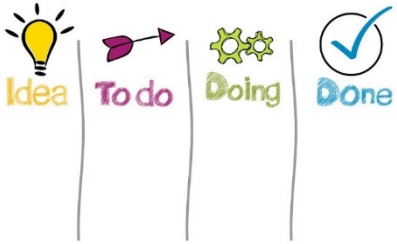
- Estimates aren‘t an assertion (just estimates)
- Definition of „Done“: everything to be done to complete the story (implementation, integration, documentation, QS, etc.)
- Person days ≠ working days: besides development, many additional activities (meetings, e-mails, etc.)
- Requirements aren‘t precise and are often adapted: precise estimates are simply not possible

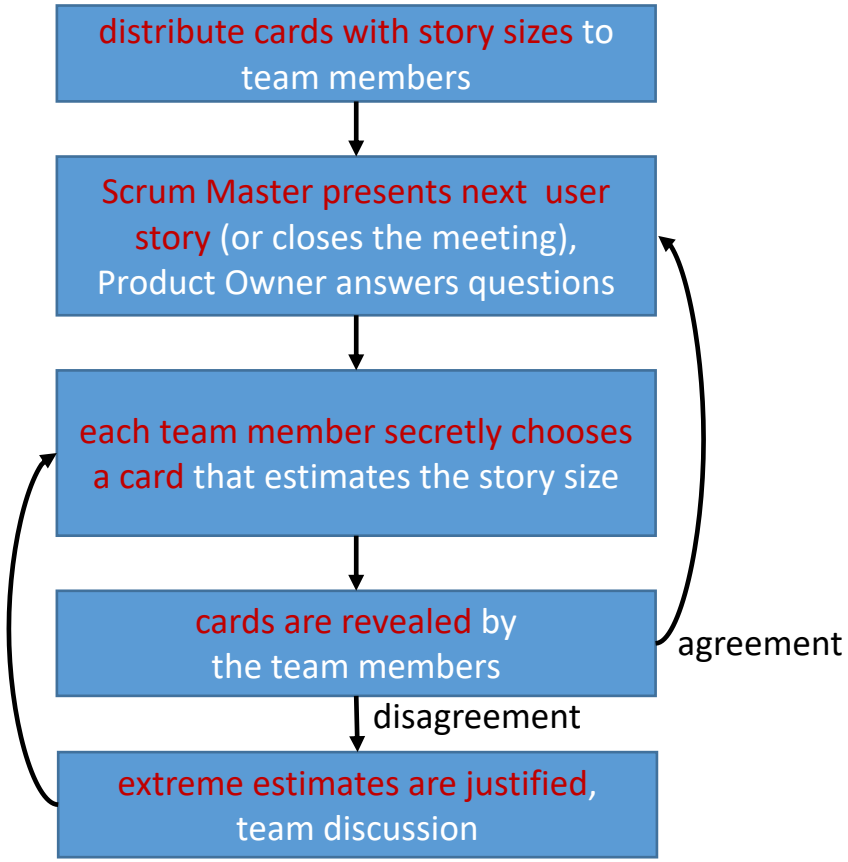
Story Points: Evaluation Scale



- Often used: exponential evaluation scale
- For example: Fibonacci sequence (1, 2, 3, 5, 8, 13, 20, …)
- Linear sequences not recommended: the larger and the less concrete a story, the lower the precision of the estimate (more aspects are unclear)
- Recommendation: user stories that are on the way to be included in a sprint, should be evaluated on a Fibonacci scale of 1..13

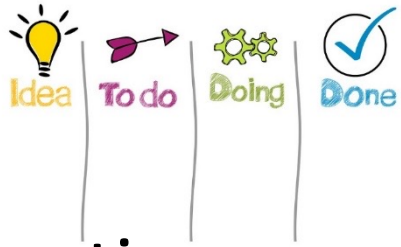
SCRUM Planning Poker



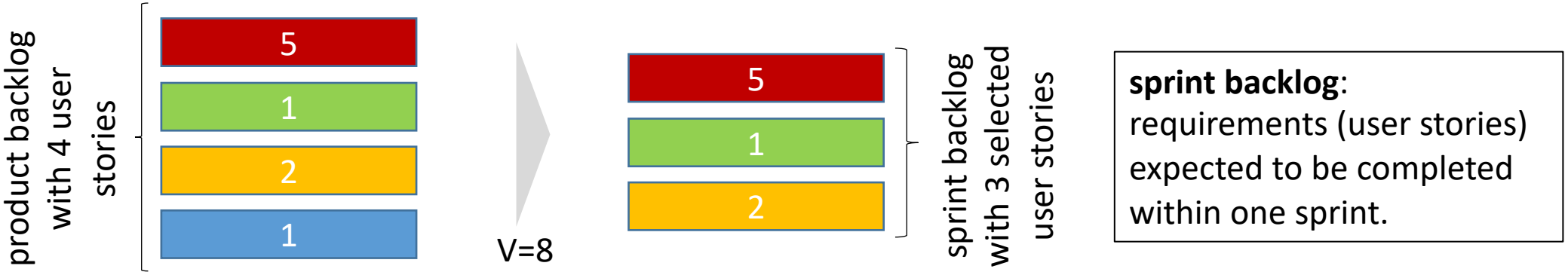


- Effort estimation during "sprint planning meeting"
- Often used method: planning poker
- Estimation cards representing "story sizes" (e.g., doubling numbers 0.5, 1,2,4, …; Fibonacci style 1,2,3,5, …)
- If agreement regarding estimates is achieved, Scrum Master records size
- Estimates are input for the prioritization of user stories

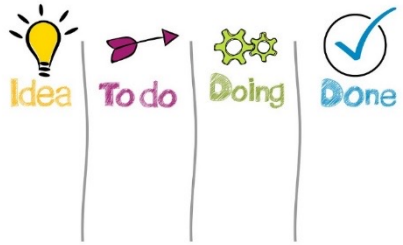
From Requirement Size to Duration



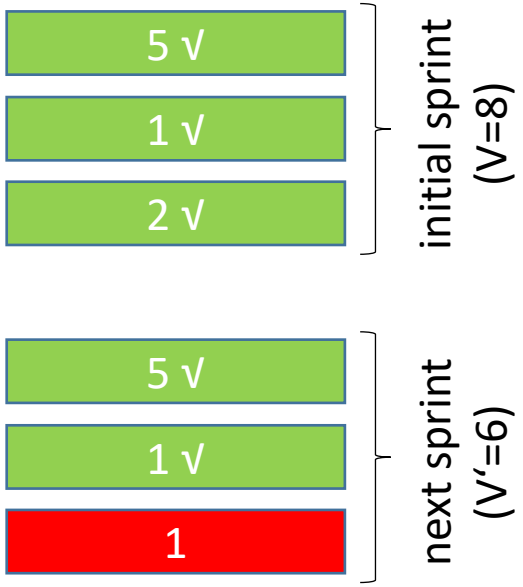
- Basis: development speed (V) of a team and sprint duration
- Velocity (V) = #completed story points per sprint
- Incomplete stories are not taken into account!
- For example: if a team‘s velocity = 14 and 28 story points are contained in the product backlog, 2 sprints are needed to develop the corresponding user stories



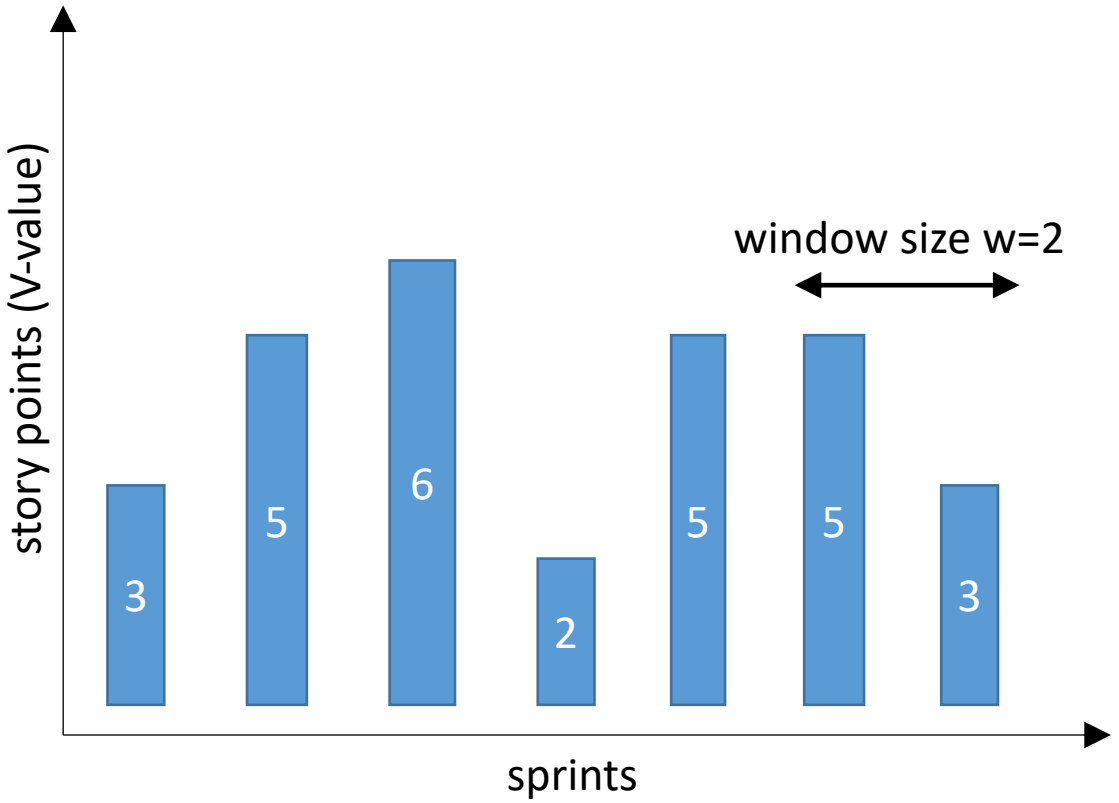
Handling Velocity



- At the beginning of a project: team takes a look at the product backlog and decides about a „realistic“ set of stories for the next sprint
- Story points of successfully completed requirements (√) represent the initial velocity V
- If V decreases in the next sprint (V‘ is the new value), the V‘ is taken as new velocity
- Should not be taken too strict: if V‘ = 0 (due to unexpected additional efforts), take V of previous sprint

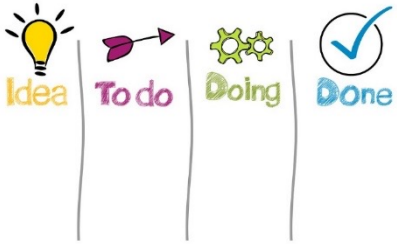


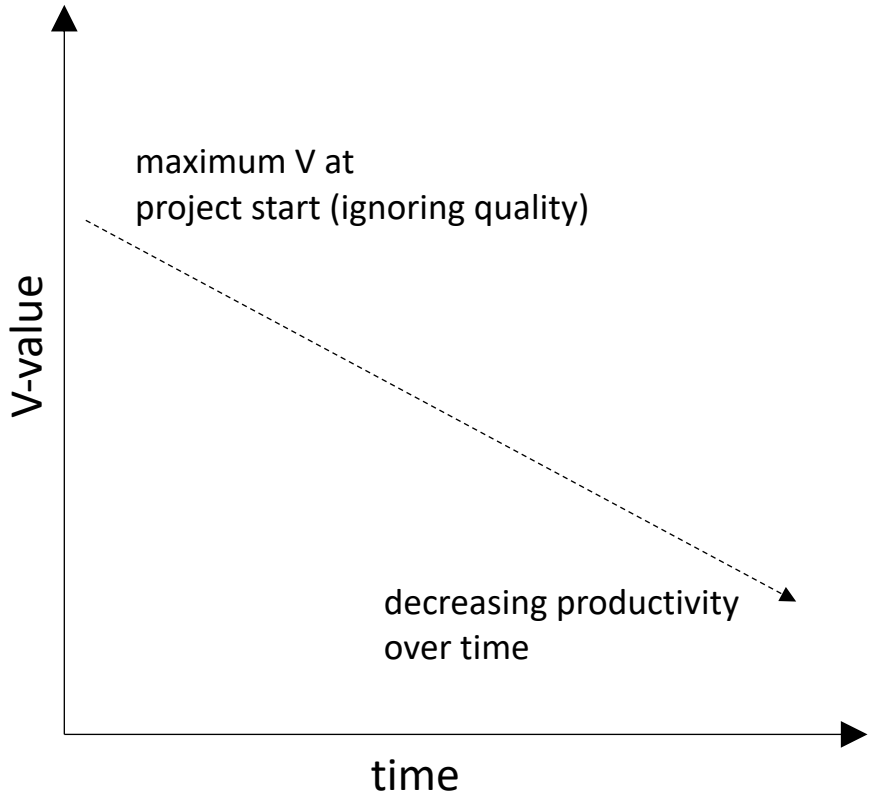
Average Velocity

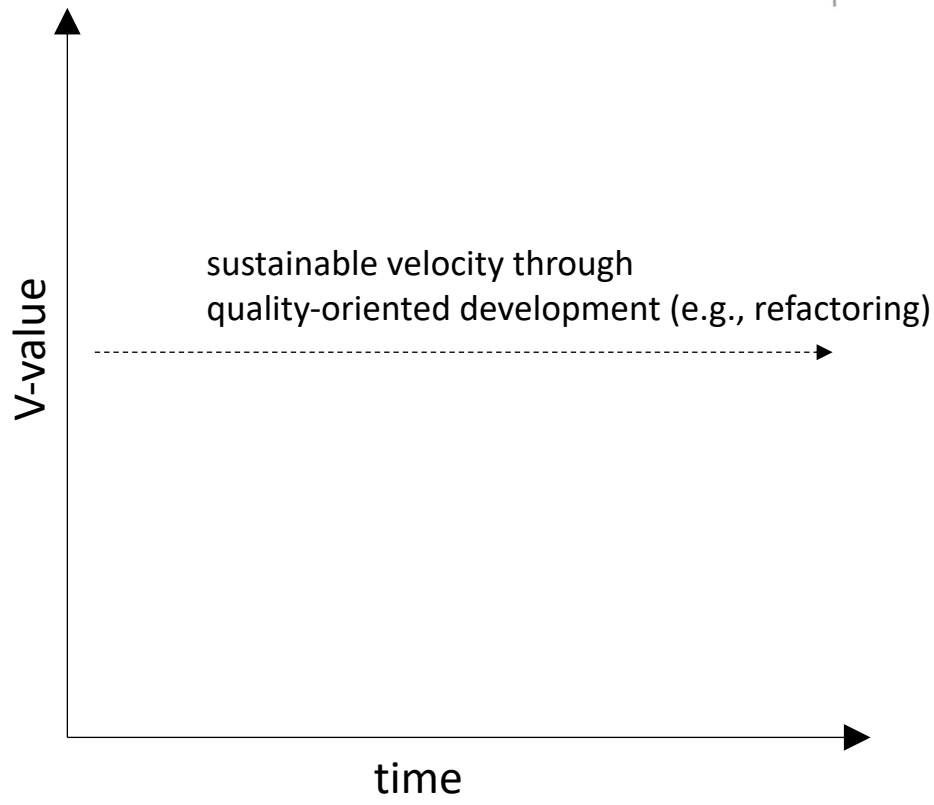


- Observation of V-values over a time period
- The more sprints, the more reliable the V-value
- Often used: Median (here: V=5)
- Compared to AVG, Median is robust w.r.t. extreme values
- Variants thereof are possible, for example, sliding window (w=2 $\rightarrow$ V=4)

Sustainable Velocity

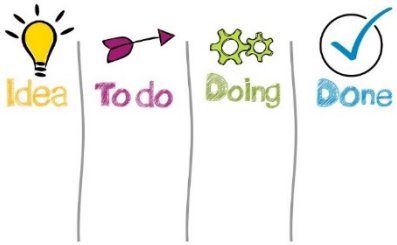






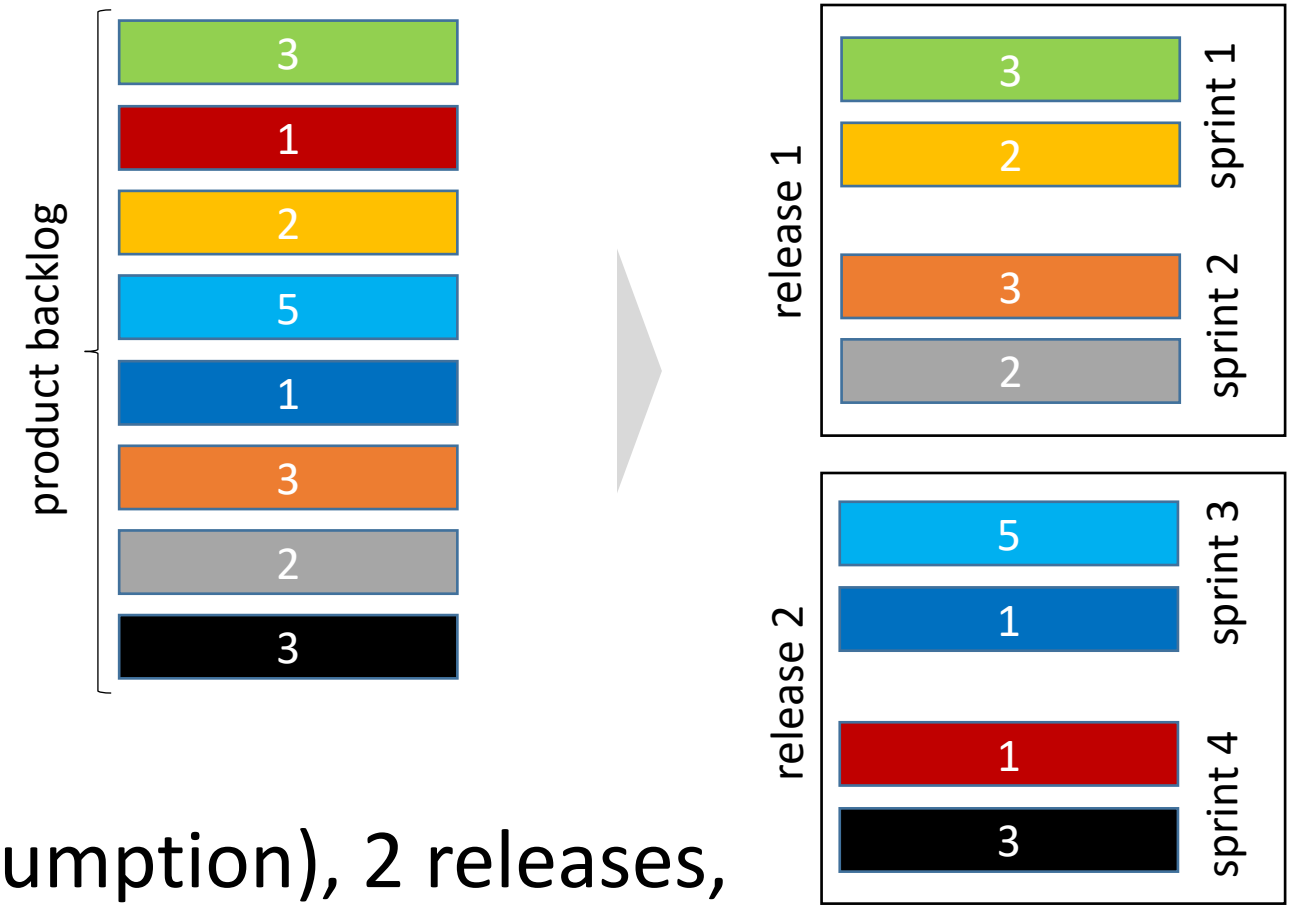
Sustainable velocity through quality-centered development

Definition of „Done“



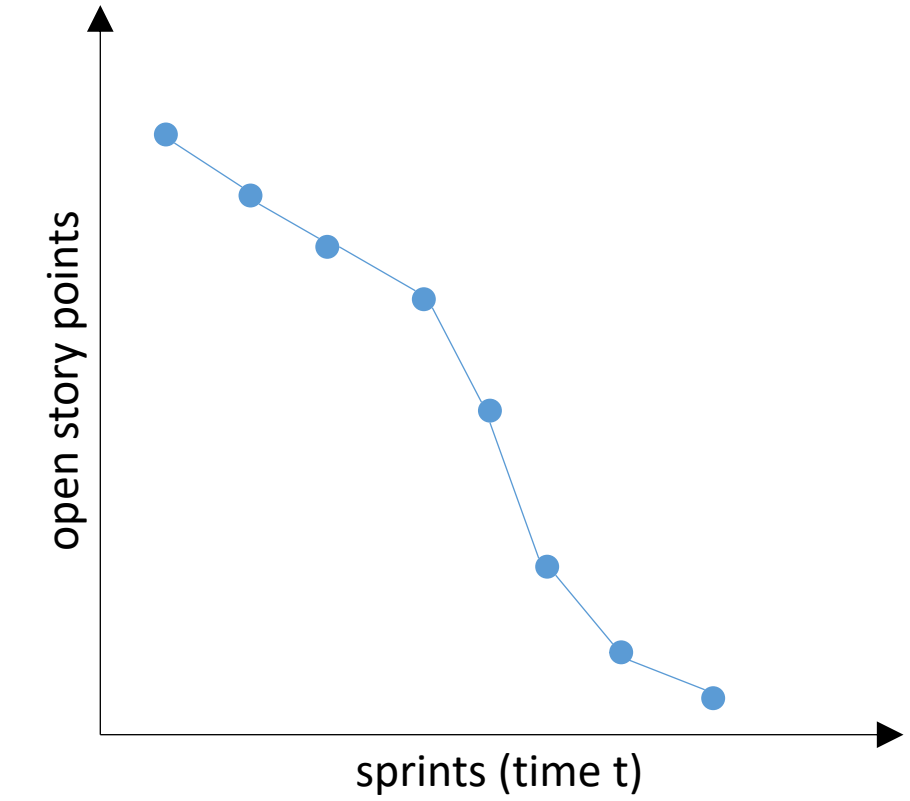
- A team „receives“ story points, although stories have not been completed in a satisfactory fashion
- Technical debt: story points have been received on credit
- Debt has to be repaid (the later this is done, the higher the overall costs and the lower the overall productivity)
- Definition of „done“: a story is completed if no related technical debts exist (code quality, refactoring, tests, …)
- Too high velocity: besides quality aspects, team members get tired and loose motivation

Relationship to Release Planning



- V=5 (assumption), 2 releases, 4 sprints

Release Burndown Chart (RBC)

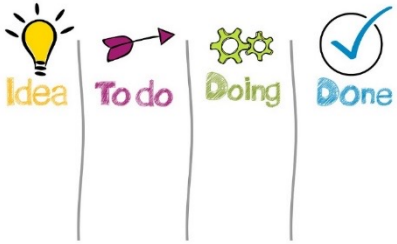


$$\text{open sprints(s)} = \text{rounded}(\frac{s}{V})$$

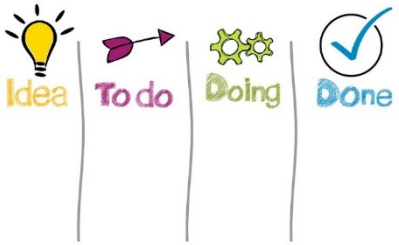
Example: still open story points s = 12, velocity V = 5 → 3 open sprints, i.e., 6 weeks given a sprint cycle of two weeks!

RBC supports an estimate of open sprint cycles until project end at a specific point of time t.

Repetition (R4)



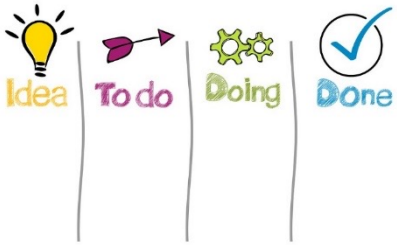
- Visit: https://checkr.tugraz.at/ (a TU Graz software).
- Login with your TU Graz student account (single sign-on supported).
- Enter the following participation code: **rJACCg** (note: you can try to answer the individual questions as often as you like!). No fixed time slots for the repetitions, **deadline for all repetitions**: **June 20th, 23:59:59**.
- Go to the category „Effort Estimation“ and answer the questions.
- Your answers will be taken into account as mentioned in the organization slides.



Thank You!

Univ.-Prof. DI Dr. Alexander Felfernig
Dr. Trang Tran
Applied Artificial Intelligence
Graz University of Technology, Austria

References



- [WIR2011] R. Wirdemann. SCRUM Mit User Stories, Hanser Verlag, 2011.