



A2DAM Stock and flow model

Revision 9

Thierry Coq, Jean-Louis Letouzey, Tom Grant, Dan Sturtevant, Jean-Pierre Fayolle

Abstract

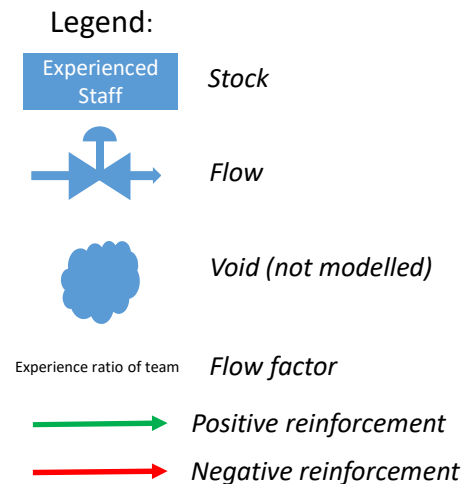
- The following stock and flow diagrams present in a clear and easily understandable way the A2DAM dynamic model.
- The key elements of a project that are affected by technical debt, or that create technical debt are modeled.
- The software production team produces software during sprint iterations taking the backlog and producing finished features that are released. The software produced may be new or updated from production. During the production process, new defects may be introduced, and new technical debt is also introduced.
- As technical debt increases it affects the team structure, discouraging experienced people on the project and reducing the ability of new staff to acquire experience. It also has effects on the production increasing defects created, reducing defect detection and further increasing technical debt.
- If the team is committed and has the ability to measure technical debt, refactoring while reducing production of new features also reduces the technical debt stock on the project counteracting the daily production of technical debt.
- This model is also available as an open-source deliverable. It is provided as-is with a Creative Commons licence and it works in the open-source OpenModelica tool, freely downloadable. The model can be tweaked and run. Feel free to run it, use it to model your own environment and provide feedback.

Abstract

- The following stock and flow diagrams present in a clear and easily understandable way the A2DAM dynamic model.
- The key elements of a project that are affected by technical debt, or that create technical debt are modeled.
- The software production team produces software during sprint iterations taking the backlog and producing finished features that are released. The software produced may be new or updated from production. During the production process, new defects may be introduced, and new technical debt is also introduced.
- As technical debt increases it affects the team structure, discouraging experienced people on the project and reducing the ability of new staff to acquire experience. It also has effects on the production increasing defects created, reducing defect detection and further increasing technical debt.
- If the team is committed and has the ability to measure technical debt, refactoring while reducing production of new features also reduces the technical debt stock on the project counteracting the daily production of technical debt.
- This model is also available as an open-source deliverable. It is provided as-is with a Creative Commons licence and it works in the open-source OpenModelica tool, freely downloadable. The model can be tweaked and run. Feel free to run it, use it to model your own environment and provide feedback.

Stock and Flow diagrams

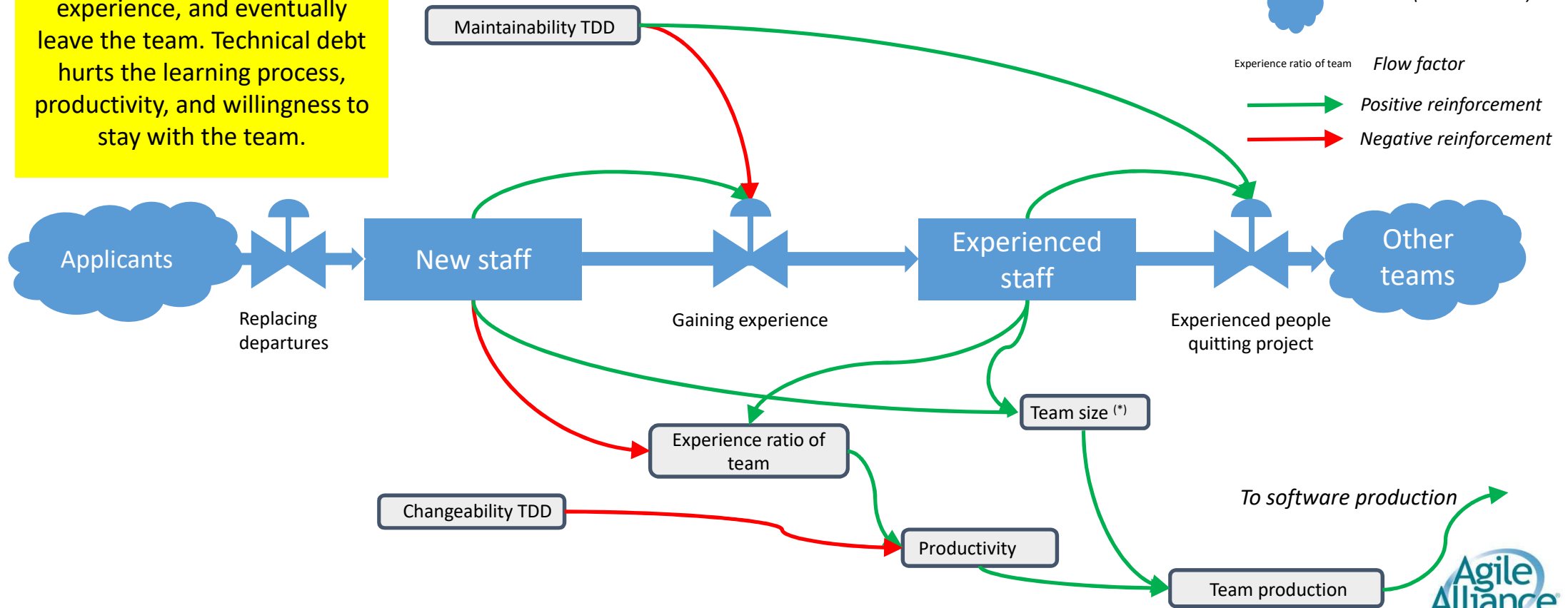
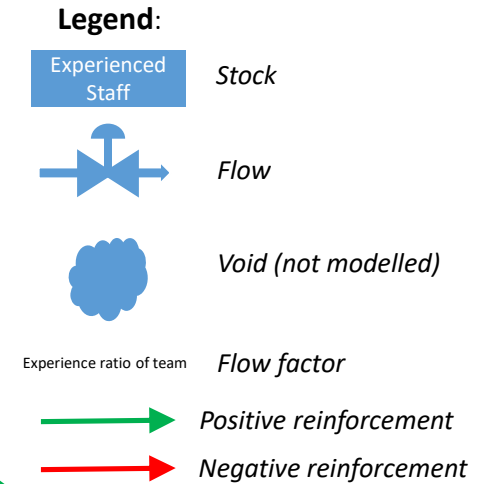
- Stocks are accumulated or consumed over time, using flows coming in and flows coming out.



- Flow rates are modified positively or negatively by flow factors that may come from outside as parameters, or may create reinforcing or balancing loops.
- SW = Software
- TD = Technical Debt
- TDD = Technical Debt Density

SF-01 : Impact of Technical Debt on team dynamics

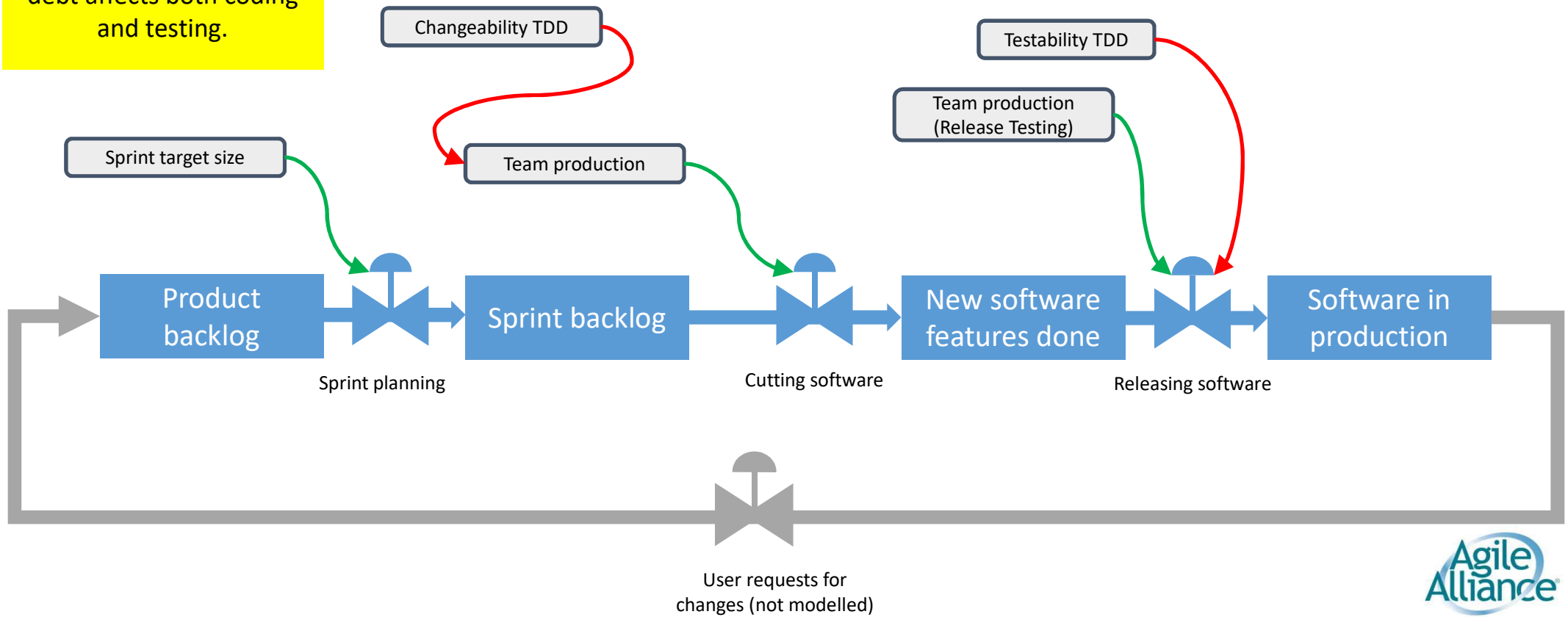
TEAM STAFFING:
 This diagram summarizes the sub-system that simulates how people join the team, get experience, and eventually leave the team. Technical debt hurts the learning process, productivity, and willingness to stay with the team.



(*) Team size is the sum of new staff and experienced staff

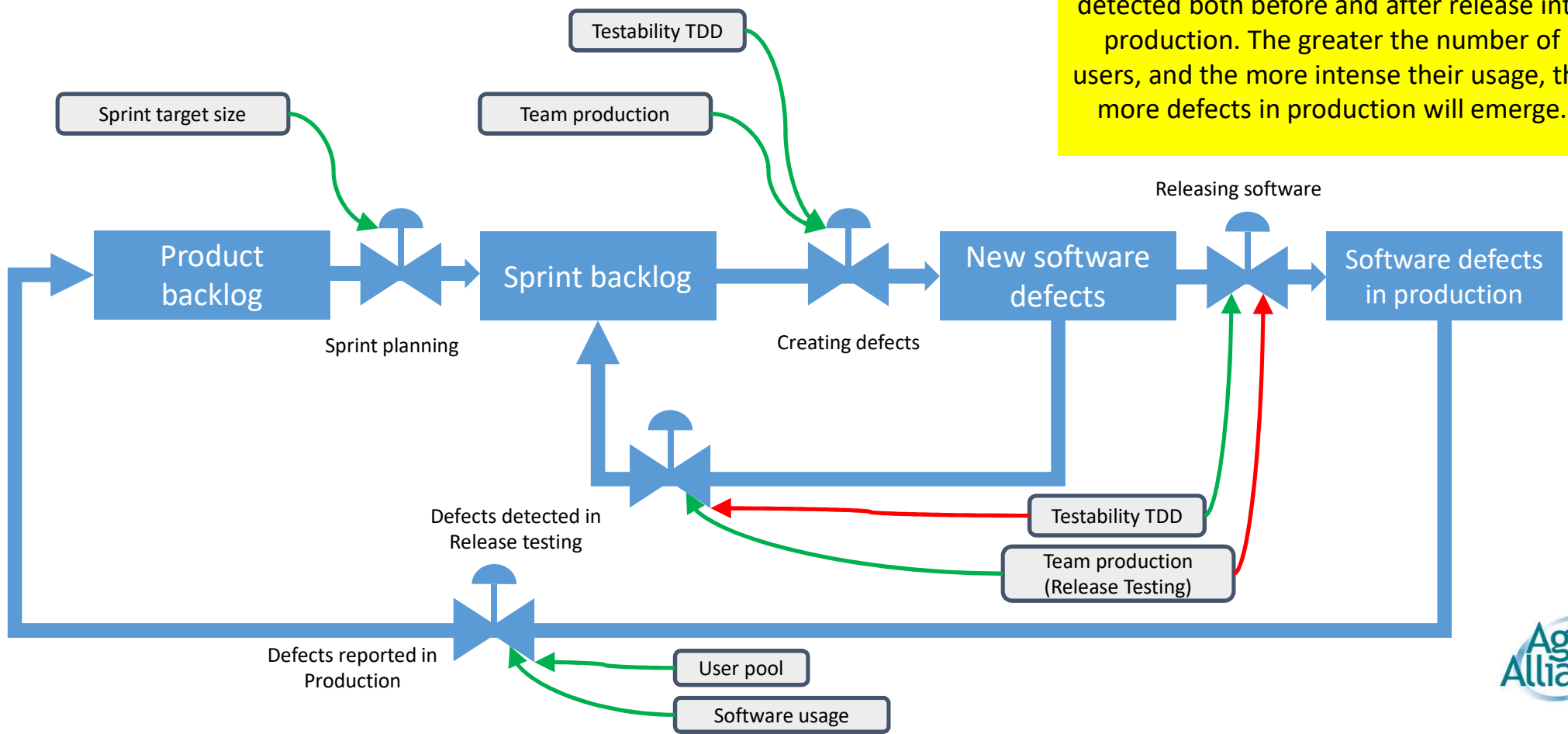
SF-02/1: Software Production and Technical Debt

SOFTWARE PRODUCTION:
This part of the model depicts the production of new software. Technical debt affects both coding and testing.



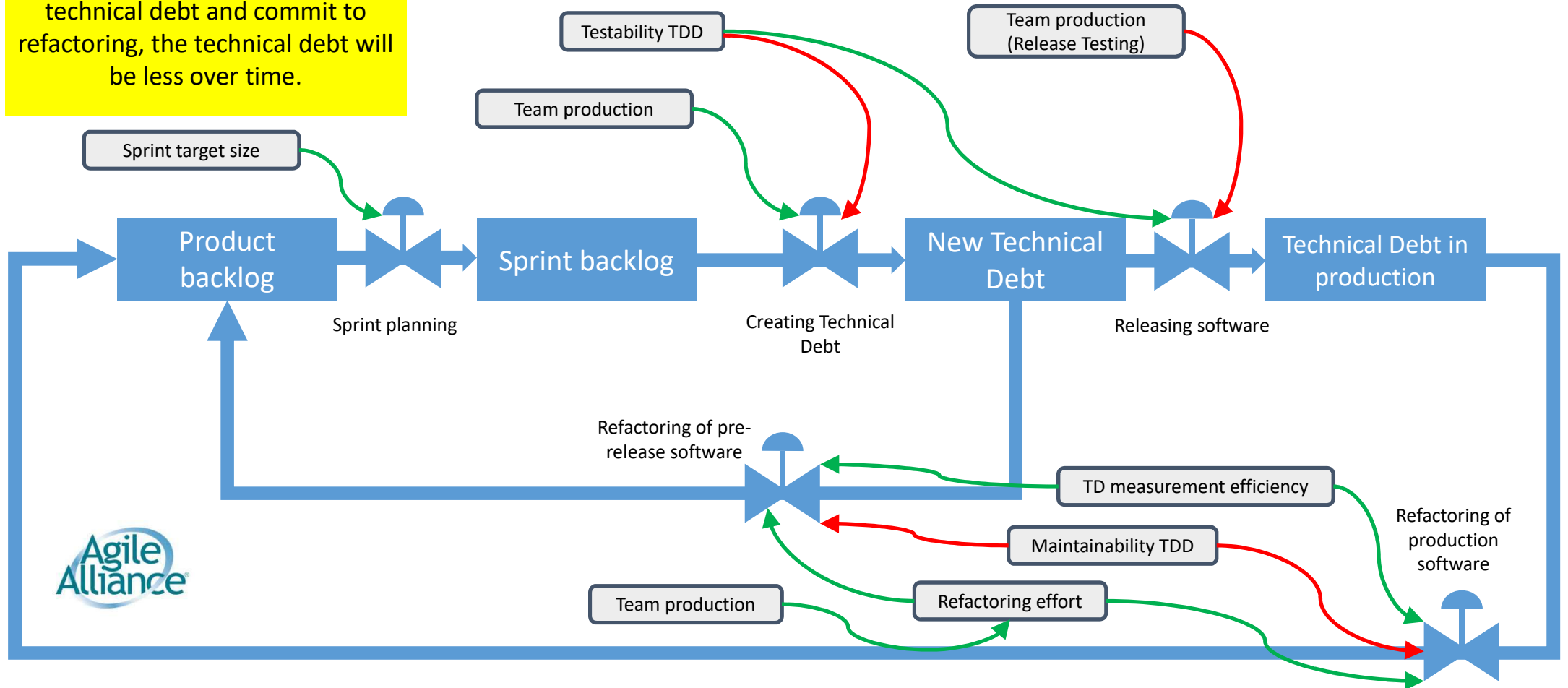
SF-02/2 : How release defects increase with technical debt

DEFECTS:
This diagram depicts how the model simulates defects, including defects detected both before and after release into production. The greater the number of users, and the more intense their usage, the more defects in production will emerge.

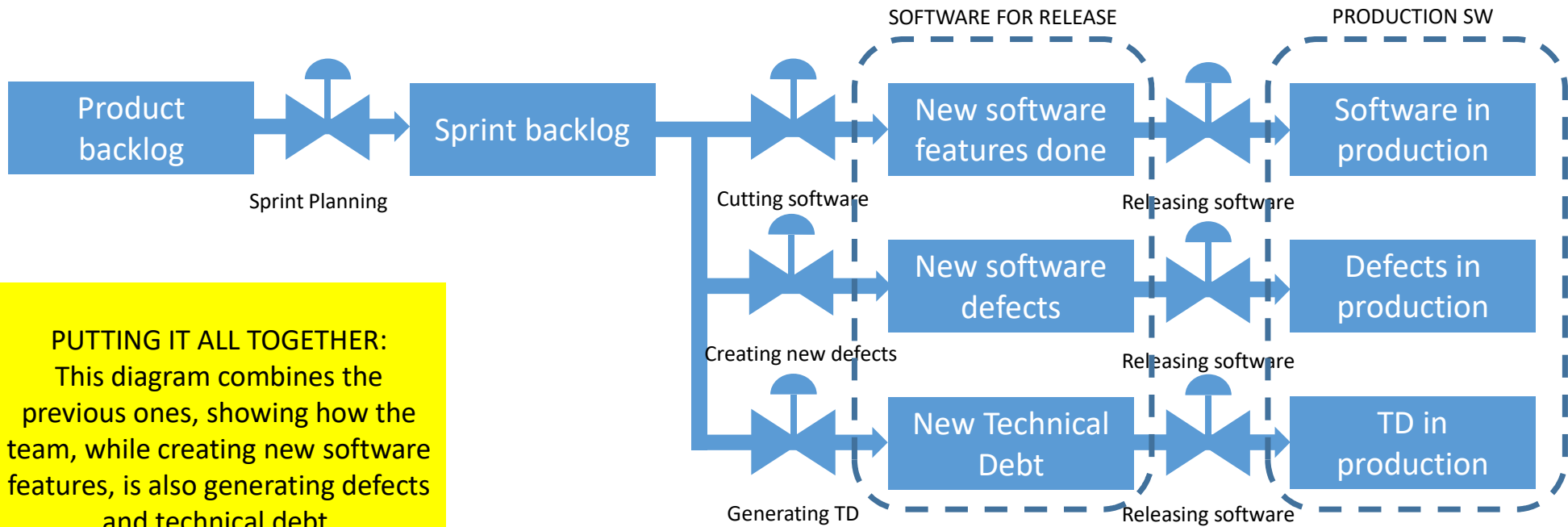


SF-02/3 : Generating Technical Debt

TECHNICAL DEBT CREATION:
This part of the model shows how the team creates technical debt. If the team decides to measure technical debt and commit to refactoring, the technical debt will be less over time.



SF-02(all) : Software Production (Features, Defects and TD together)



PUTTING IT ALL TOGETHER:
This diagram combines the previous ones, showing how the team, while creating new software features, is also generating defects and technical debt.

Definitions of Technical Debt, based on the A2DAM Dynamic model

From this model, we can deduce a definition of technical debt by its effect on projects:

- Testability (and reliability) Technical Debt:
 - Reduces the ability to produce correct software and increases the number of defects in software
 - Reduces the ability of detect defects in pre-release software
- Changeability TD:
 - Reduces productivity of the software production team
- Maintainability TD:
 - Hurts the learning process
 - Hurst the willingness to stay with the team
 - Reduces the effectiveness of refactoring activities