GÉANT
Networks • Services • People

Designing a Maturity Model for a Distributed Software Organization

Experience report

Bartosz Walter and Marcin Wolski
Poznan Supercomputing and Networking Center, Poland
Andrijana Todosijevic
AMRES, Serbia
Zarko Stanisavljevic
University of Belgrade, Serbia

EuroSPI 2019, Edynburgh, 18-20 of Sep 2019

www.geant.org

Maturity in software development

- What does *maturity* mean in context of software development?
  - Possible meanings: replicability, optimization, ability to reflection, evaluation and improvement
- What are directions of maturity?
  - **Process-oriented**: self-improving process leads to the better team performance
  - **Product-focused**: make only improvements that directly help your product
  - **Engineering**: search for better tools, frameworks, infrastructure
- Why do we need *models* of maturity?
  - To capture the current state
  - To define a roadmap for improvement
  - To make a performance evaluation

Overview of existing maturity models

- Maturity in software development is usually defined in terms of the Capability Maturity Model Integration for development – CMMI-DEV or the international standard ISO/IEC 15504 (now revised by ISO/IEC 33002) [1]
- Mettler and Rohner [2] identified a list of 135 different maturity models related to the discipline of information systems
- In the field of software engineering, von Wangenheim et al.[3] identified 52 software process capability/maturity models (SPCMM)
- Fontana et al. [4] found and analyzed 14 models combining CMMI-DEV with agile methods

Generic vs. custom maturity model

- **Generic maturity models** – for a „typical” organization
  - Mandate the recognition and implementation of certain activities defines by the model
  - Example: CMMI, TMM, ISO/IEC 15504 (SPICE) are commonly used in practice

- **Custom maturity models** – for organizations with specific needs, constraints, environments
  - Usually they refer to common values or principles (e.g. agility)
  - Example: Progressive Outcomes [5]

- **Observation**: maturity is a balance between optimal performance and the ability to adapt. Some adjustments can be made within a generic framework, but more extensive changes require defining a new model (based on existing models)

Our case: GEANT

- GEANT is a research and innovation organization, built upon of federation of NRENs
- There are around 30 software project in the GEANT portfolio, with ~300 individuals from GEANT, who contribute to the software development.
- Features specific to GEANT:
  - **Geographical dispersion** and distribution of teams
  - Mostly **small teams** (2-5 members), preferring iterative software development
  - Emphasis **on innovation and exploration** of concepts
  - **Freedom of teams** in choosing tools, methodology, practices etc.
- Is GEANT so unique that it needs a custom maturity model?

Software Maturity Model for GEANT (revised)

Requirements Engineering

- Identification and management of stakeholders
- Eliciting needs of stakeholders
- Analysis of requirements and their dependencies
- Traceability of requirements
- Validation of requirements

Design and Implementation

Target Area

Specific goal

...

Team Organization

Target Area

Specific goal

...

Quality Assurance

Target Area

Specific goal

...

Software Maintainance

Target Area

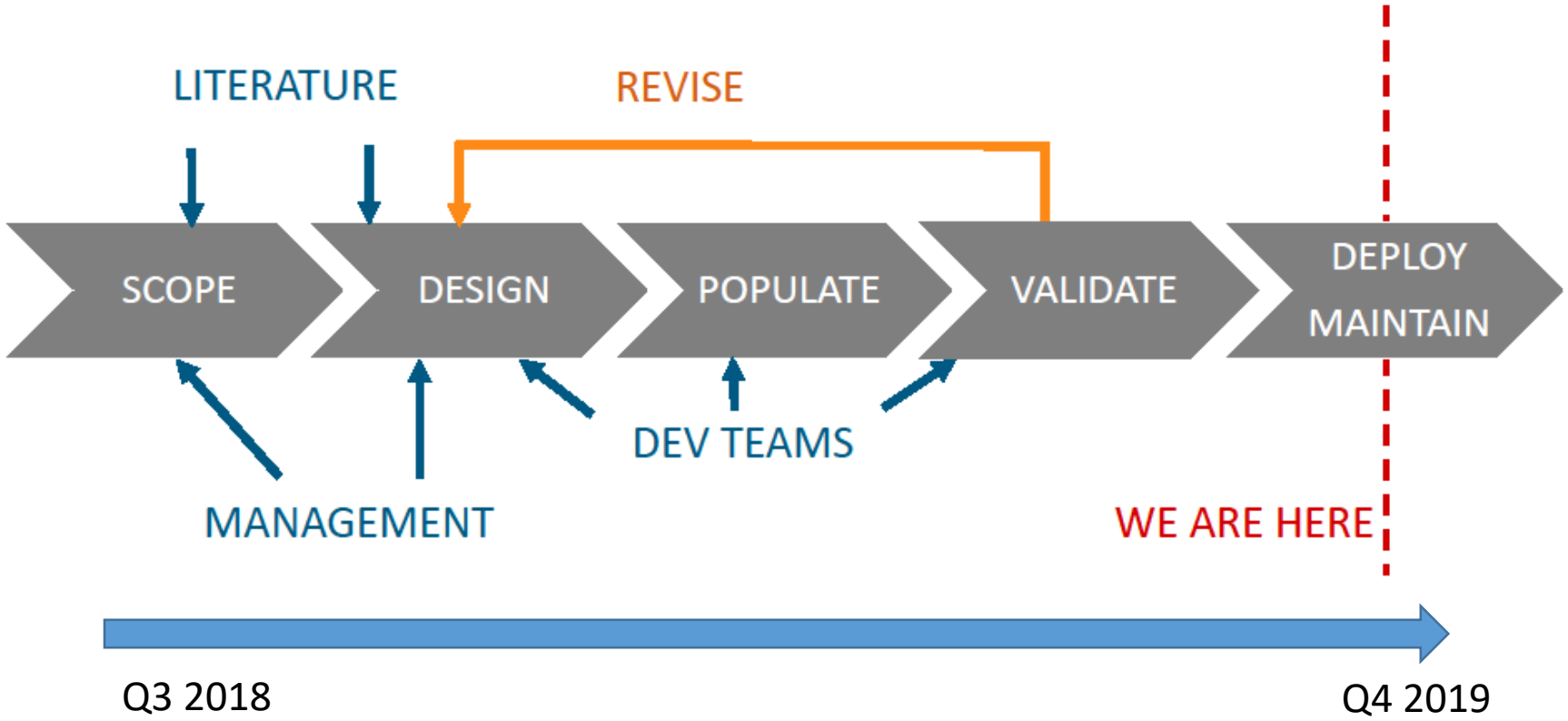
Specific goal

...

Based on CMMI Level 2

GEANT specific

Process of defining the model



Observations concerning the process of defining the model

1. Identify and involve relevant stakeholders
2. Identify and clearly present the objectives
3. Capture contextual factors
4. Early and frequently validate the model with stakeholders
5. Refine the model iteratively

Identify and involve relevant stakeholders

1. Project has **different stakeholders** with **diverse perspectives and needs**
2. Stakoholders should be identified and involved, to **avoid tensions**
3. Actively **look for new relevant stakeholders**
4. Initial groups of stakoholders: subject teams, senior management, process-quality teams

Identify and clearly present the objectives

1. The objectives for the MM should be **transparently communicated to all stakeholders**
2. Every **objective has to be justified**
3. Discrepancies should be **negotiated and reconciled**
4. Partition the objectives into **primary and secondary**; focus on the primary, not forgetting about the secondary. Get the approval for this partition from stakeholders

Capture contextual factors

1. Identify **factors that make your organization unique** or **may change in time**
2. Propose adequate methods for addressing the identified factors

Early and frequently validate the model with stakeholders

1. Stakeholders will contribute to the model, if it would not be contradictory with their objectives
2. A communication strategy should include **early and frequent consultations with stakeholders**

Refine model iteratively

1. Add new elements/objectives in **increments**
2. **Balance the size of an increment** (too large to accomodate vs. too frequent to manage)
3. **Get approval** from stakeholders

Summary

- GN-SMM delivers a framework for improving the processes in organization (more emphasis on supporting the teams, much less – on their evaluation)
- Maturity Models need to embrace the needs and constraints of various stakeholders
- The identified guidelines for the process are compliant with SPI Manifesto
- Next steps:
  - evaluate and revise the SMM, based on interviews with software teams
  - deploy the model in GEANT and monitor its performance

GÉANT
Networks • Services • People

Thank you

Any questions?

www.geant.org

© GÉANT Association on behalf of the GN4 Phase 3 project (GN4-3).
The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 856726 (GN4-3).

Selected references

1. G.A. Garcia-Mireles, M. Angeles Moraga, and F. Garcia. Development of maturity models: a systematic literature review. In IET Seminar Digest, volume 2012, pages 279-283, 2012.
2. Mettler, Tobias, and Peter Rohner. "Situational maturity models as instrumental artifacts for organizational design." *Proceedings of the 4th international conference on design science research in information systems and technology*. ACM, 2009.
3. von Wangenheim, Christiane Gresse, et al. "Systematic literature review of software process capability/maturity models." *Proceedings of International Conference on Software Process Improvement and Capabity Determination (SPICE), Pisa, Italy*. 2010.
4. Rafaela Mantovani Fontana, Regina Albuquerque, Rosmar Luz, Ana Carolina Moises, Andreia Malucelli, and Sheila Reinehr. Maturity models for agile software development: What are they? In European Conference on Software Process Im- provement, pages 3{14. Springer, 2018.
5. Rafaela Mantovani Fontana, Victor Meyer, Sheila Reinehr, and Andreia Malucelli. Progressive Outcomes: A framework for maturing in agile software development. Journal of Systems and Software, 102:88{108, 2015.
6. N. Ramasubbu, M. S. Krishnan, and P. Kompalli. Leveraging global resources: a process maturity framework for managing distributed development. IEEE Software, 22(3):80{86, May 2005.
7. Alain April and Alain Abran. A software maintenance maturity model (s3m): Measurement practices at maturity levels 3 and 4. Electronic Notes in Theoretical Computer Science, 233:73 (SQM 2008).
8. Tonia De Bruin, Ronald Freeze, Uday Kaulkarni, and Michael Rosemann. Understandingthe main phases of developing a maturity assessment model. In B Campbell, J Underwood, and D Bunker, editors, Australasian Conference on Information Systems (ACIS), pages 8{19,
9. Jens Poppelbu and Maximilian Roglinger. What makes a useful maturity model? a framework of general design principles for maturity models and its demonstration in business process management. In ECIS, 2011.