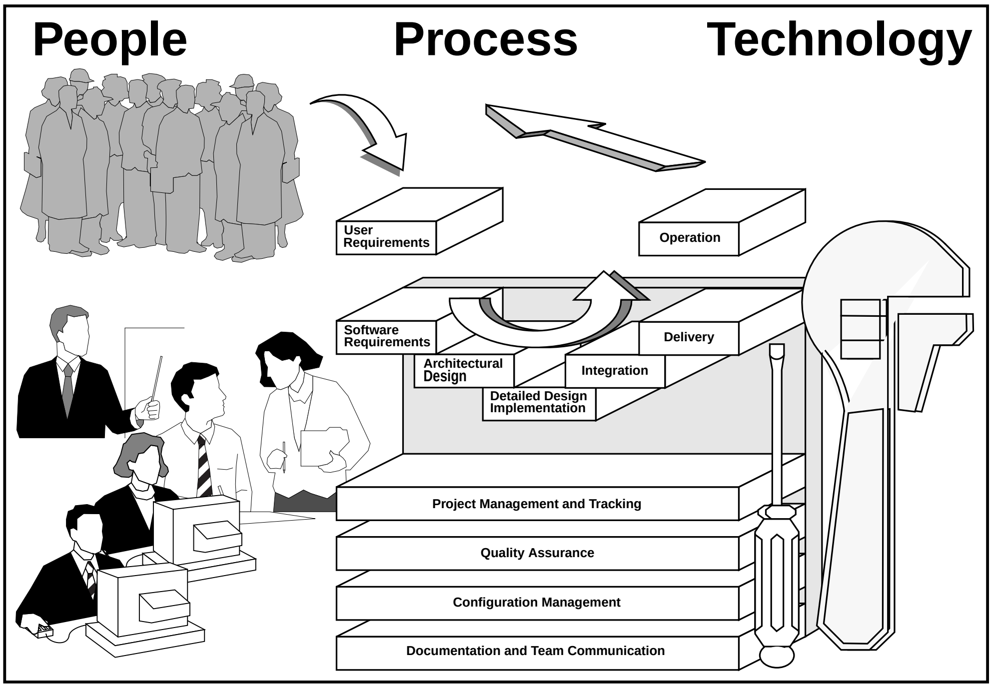
SOFTWARE EVOLUTION - TRACK INTRODUCTION

Arash Khodabandeh
Information, Process and Technology Group
CERN, Geneva, Switzerland

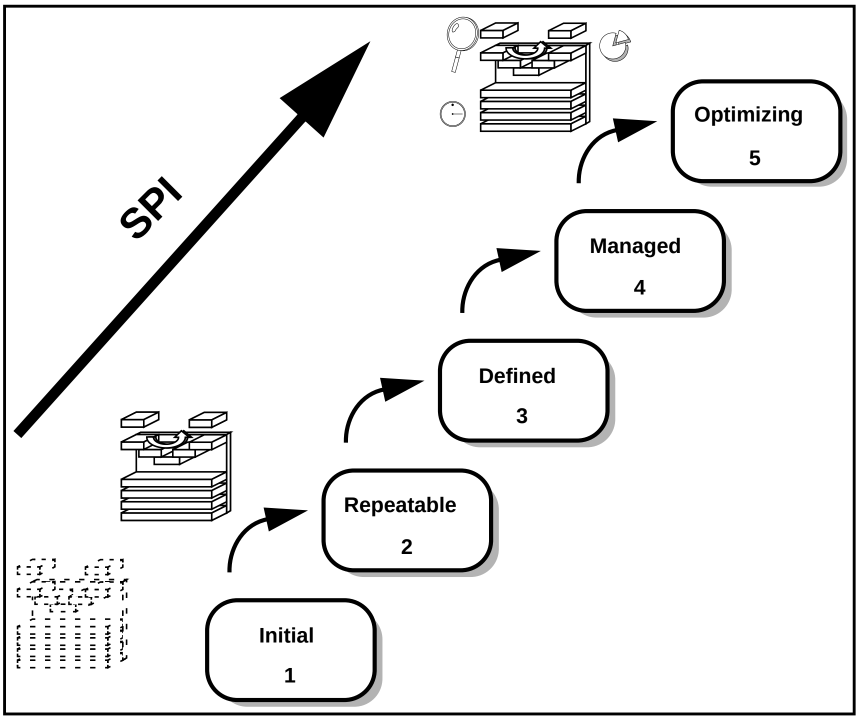
The production of software is a labor intensive activity, especially in the field of High Energy Physics, where the complexity of current and future experiments require large software projects.

For most of the scientists and engineers involved in software production, the business is science or engineering, not computing. As software scope continues to grow so does the feeling that its development and maintenance are out of control. The situation is made even worse by a lack of software engineers and from an uneven software culture. Better organization and control of software production are clearly needed in order to face the challenge of developing new software for the LHC and of maintaining software for previous experiments.



To be able to control the production of software it is essential to improve (a) the knowledge of the PEOPLE involved, (b) the organization and improvement of the software development PROCESS and (c) the TECHNOLOGY used in the various aspects of this activity. The goals are better systems at lower cost, and of better quality.

The process is the set of orderly actions to be performed to produce the software throughout the life cycle. The quality of a process can be measured in terms of maturity against a recognized framework. A reference framework is the Capability Maturity Model (CMM) proposed by the Software Engineering Institute (SEI). This model consists of five levels and the most difficult step is to move from level 1 to level 2 because of all the management procedures and activities that have to be put in place. As an organization moves up the maturity levels, management visibility on the software process improves, estimates become more accurate, schedules are met more precisely and the time required to produce a software system shortens. This progression in maturity levels is also called Software Process Improvement (SPI).



Except for level 1 (where all organizations start), each maturity level is decomposed into several Key Process Areas (KPA) that indicate the actions an organization should focus on to improve its software process and reach the corresponding level. The Software Evolution track uses the level 2 as a file rouge. Every lecture teaches one level 2 KPAs:

- User Requirements by G.Kellner (CERN IPT Group)
- Project Management and Tracking by A.Aimar (CERN IPT Group)
- Configuration Management by B.Jacobsen (UC Berkeley) and D.Johnson (Univ. Colorado)
- Quality through Software Metrics by A.Khodabandeh (CERN IPT Group)

The track also contains one lecture on documentation, an important aspect of software engineering too often overlooked:

- Software Documentation by A.Aimar (CERN IPT Group)

Two case studies illustrate the adequacy of the presented concepts to our environment by showing how they have been put into practice in HEP experiments:

- BaBar Case Study by B.Jacobsen (UC Berkeley)
- OPAL Online - A Case Study by K.Ackerstaff (CERN OPAL)

The track consists of 8 hours of lectures, 11 hours of exercises and wrap-up, and 4 hours of case studies.

REFERENCES

[1] ESA, “Software Engineering Standards”, Prentice-Hall, ISBN: 0-13-106568-8

[2] ESA, “Software Engineering Guides”, Prentice-Hall, ISBN: 0-13-449281-1

[3] W.S.Humphrey, “Managing the Software Process”, Addison-Wesley, 1990, ISBN: 0-201-18095-2

[4] W.S.Humphrey, “A Discipline for Software Engineering”, Addison-Wesley, 1995, ISBN: 0-201-54610-8

[5] W.S.Humphrey, “Introduction to the Personal Software Process”, Addison-Wesley, 1997, ISBN: 0-201-54809-7

[6] W.S.Humphrey, “CASE planning and Software Process”, CMU-SEI-89-TR-26, May 1989, http://www.sei.cmu.edu/products/publications/89.reports/89.tr.026.html

[7] V.Stenning, “On the Role of an Environment”, *9th International Conference on Software Engineering*, Montery, California, March 30, 1987

[8] W.E.Deming, “Quality, productivity and Competitive Position”, MIT 1982

[9] SEI, “Capability Maturity Model for Software”, V1.1. CMU/SEI-93-TR-24, 1993

[10] SEI, “Key Practices of the Capability Maturity Model”, V1.1. CMU/SEI-93-TR-25, 1993

[11] J.Herbsleb, A.Carleton, J.Rozum J.Siegel, D.Zubrow, “Benefits of CMM-Based Software Process Improvement: Initial results”, CMU/SEI-94-TR-13, 1994

[12] SEI, “Software Quality Assurance”, SW-CMM v2 Draft A, Oct 96, http://www.sei.cmu.edu/technology/cmm/draft-a/a25qa.html

[13] B.J.Brown, “Assurance of Software Quality”, SEI-CM-7-1.1, Jul 87, http://www.sei.cmu.edu/products/publications/cms/cm.007.html