

Software Product Engineering An Overview

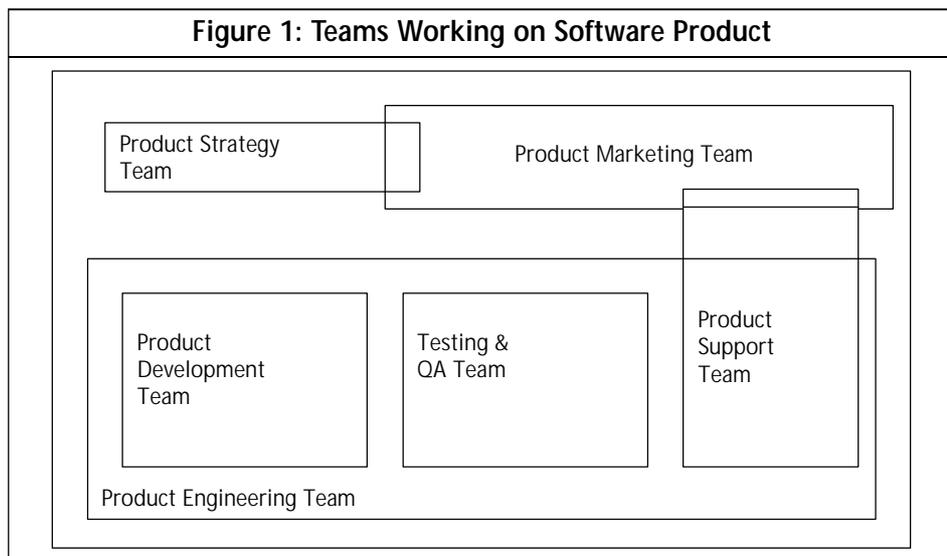
G P Sudhakar

This article gives an overview of the software product engineering. There are few organizations in the world such as Microsoft, ORACLE, SUN, HP and SAP, which have championed the product engineering process. It explains the typical steps involved in software product engineering such as requirements elicitation, analysis, different stages of designing, coding, building and packaging, different test strategies, different levels of testing and release management. Finally, it also explains the project management and product development teams.

1. Introduction

There are many product-oriented companies in the world such as Microsoft, HP, Oracle, SAP, Sun Microsystems, etc. The success of any software product depends on how it is being developed and engineered. All these companies are having strong engineering departments and teams working on their products. This article explains the typical software product engineering process in any software product-development company.

Basically, software product management is an interdisciplinary function, which integrates functions of engineering, marketing and sales. To any software product there are product strategy team, marketing and sales team, product support team, product development team, testing team and QA team. Broadly speaking, product development, product testing, QA and product support come under product engineering department. It varies from organization to organization and expressed pictorially in the Figure 1.



2. Requirements Gathering and Engineering

Business requirements elicitation is the first step in product development. According to Bashar & Steve (2000), the requirements engineering includes the following steps:

- Eliciting requirements;
- Modeling and Analyzing requirements;
- Communication of requirements;
- Agreeing requirements; and
- Evolving requirements.

Basically, the collected requirements are analyzed and communicated to stakeholders. The acceptance of stakeholders is necessary to proceed to the design stage of the product. The project manager or the product manager of the engineering should understand the needs, wants and expectations of the stakeholders.

Requirements elicitation includes collecting the requirements with the help of traditional techniques such as surveys, questionnaires, and analysis of existing documents, interviews (both structured and unstructured interviews), brainstorming and focused groups.

According to Bashar & Steve (2000), the gathered requirements should undergo the modeling methods such as enterprise modeling, data modeling, behavioral modeling, and domain modeling. Once modeling and analysis of requirements are completed, then the requirements are communicated to the stakeholders. A better tool to trace the requirements is the requirements traceability matrix. Over the course of product, the product management needs to manage the change of requirements. This can be done through Change Control Boards (CCB) established in the organization. Requirements are to be validated before going for agreement by all the parties involved in the product. Once requirements elicitation is over we can proceed for analysis and design stages.

3. Product Analysis and Design

Once technical requirements are gathered then they should undergo technical analysis and design stages.

3.1 Requirements Analysis

Many techniques are used during the analysis stage. Usually, Structured Systems Analysis (SSA) or Object-Oriented Analysis (OOA) techniques can be applied to the systems analysis. This OOA includes finding out the abstractions such as classes and objects involved in the problem specification of the product. If we follow the SSA, we will be identifying the entities in the system in this phase.

3.2 High Level Design

In high level design stage using Object Oriented Design (OOD) we find the relationships among the classes such as generalization, specialization, aggregation and composition relationships. If we follow the Structured Systems Design (SSD) techniques we will be using functional decomposition diagrams, data flow diagrams, etc. to design the software systems. In SSD, the relationships among the Entities are identified in the stage of high-level design. Basically OOD and SSD are widely used *design* techniques in the industry to design software systems and products. The object oriented design notations include Booth notation, OMT (Object Modeling Techniques) and the modern UML (Unified Modeling Language). By using these notations one can design software systems.

3.3 Low Level Design

In low-level design stage, OOAD (Object Oriented Analysis and Design) helps to identify the methods of the classes and their relationships. Which the caller and called methods will be identified in this stage. SSAD (Structured Systems Analysis and Design) helps to identify the functions which calling and called. Which functions are using which data structures such as structures and Unions can be identified in low-level design stage.

3.4 Using Design Tools

Many design tools are available in the market for the purpose of software design. The Rational Rose is widely used tool. These days the design tools and Rapid Application Development (RAD) tools are available which can even generate the templates of the source code by taking the classes and their relationships as the input. These design tools are useful to generate the diagrams such as Class diagrams, Object diagrams, Interaction diagrams, Sequence diagrams, Collaboration diagrams, State chart diagrams, and Deployment diagrams, etc. All these diagrams can be included in the design documents. Companies such as NEC, prepares both high level and low level design documents for their software projects.

4. Writing Efficient Source Code

Once the classes, methods and functions are identified in the design stage then they can proceed to the coding stage of the software product. Software coding

can be done by using Integrated Development Environments (IDE) and Editors. Usually IDEs provide Compilers, Debuggers, Assemblers and Interpreters to work on the source code. Using these tools one can prepare the application libraries or the executables to deploy on to the specific computers. The widely used programming languages include C, C++, Java, COBOL, Ada and SmallTalk. Now even framework such as Microsoft Dot Net is also available for the purpose of software product development.

5. Role of Open Source Software

The open source software like Linux has got significant importance in product development. CVS is another open source version control system, which is available for the purpose of software configuration management.

According to Diomidis & Clemens (2004), there are around 30,000 open source projects registered with <http://freshmeat.net>, 70,000 open source projects registered with <http://sourceforge.net> and around 5,400 Perl modules with the Comprehensive Perl Archive Network (www.cpan.com). Any Software Product, being developed in the industry, can make use of these available open source software projects and can extend or reuse them based on their needs. Hence, open source software has got significant importance in software product development. It may reduce the development time if we find the needed components or libraries in the open source projects.

6. Product Testing Strategies

Once the product code development is over, one needs to build and package the product. The packaged product goes for system testing. Product testing strategies include black box (functional) testing and white box (structural) testing and statistical testing. The test strategies have different levels of coverage such as:

- Statement Coverage;
- Branch Coverage; and
- Path Coverage.

Every software product should have test plan. Based on the level of testing unit, integration, system, and acceptance test cases needs to be prepared and documented.

6.1 Product Installation

Usually, the product building happens in the SCM (Software Configuration Management) tool environment. Once building is done, the built libraries and the executables need to be packaged by using packaging tools such as InstallShield. The packaged product is to be installed on the specified computer. Here engineers test the validity of the Installation and check out for the valid license. The engineer even examines the specific directories for proper installation. Once Installation of the product is done then it needs to be configured.

6.2 Product Configuration

Some products need configuration before invoking the application. The test engineer or the Installation engineer configures the product by doing needed initialization of the configuration files on the specific computer. This configuration is very much required if the product is going to do any read or write operations on the computer. Once the configuration is done, then the product is ready for system test or work (if it is already tested product).

6.3 Product Testing

The product testing can be done at the following levels:

- 1) Unit Testing,
- 2) Integration Testing,
- 3) System Testing, and
- 4) User Acceptance Testing.

Usually, the programmer, who developed the source code does *Unit Testing*. For doing unit testing, source code, unit test cases and test program should be available. It is better to document the test cases and test procedures. It is a best practice to review the test cases as well in addition to the review of requirements specifications and design documents.

Integration Test tests the different modules of the system and their interaction. It involves testing of two or more than two related classes and methods. Test Engineers prepare these test cases. In some cases it may involve test on multiple machines as well.

System Test starts with proper installation and configuration of the product on the test machines at the performing organization. It covers the entire product testing by using all the interfaces. Some of the products may even have hundreds of system test cases.

User Acceptance Test (UAT) is done at Customer location, mostly. The customer prepares UAT test cases. Based on the success of these test cases only customer accepts the product. Once the customer accepts the product, then it enters into maintenance stage of the product. It is the job of support teams, to support the product.

6.4 Test Management Tools

There are many test management tools such as Test Director available in the market for test management purposes. By using these tools, one can save the test cases; store the execution status of the test cases, and save different rounds of testing. Reports also can be generated by using these tools.

6.5 Test Automation

Testing is of two types. They are manual testing and automated testing. Manual testing is time consuming process. It should be done by test engineer, who tests one test case at a time whereas test automation involves running the multiple test cases at a time in sequence by a test program. Here test automation tools such as Win Runner, Load Runner and Silk Test can be used. If a bug is found during the testing, it can be tracked by using bug-tracking tools.

6.6 Ensuring the Quality

Quality Assurance is done through periodic reviews of the deliverables during the product development process. For example, reviewing requirements specifications, design documents, etc.

Quality Control is done through finding defects in the product through different levels of testing such as unit, integration and systems testing and fixing them.

The objective of both Quality Assurance and Quality Control is to deliver good quality product to the customer and to improve the product development process in the performing organization.

7. Release Management

Usually, every product will have different releases with different features. This release management uses version control tools extensively. By using the version control tool, we can build the specific version of the product in the SCM environment.

7.1 Building

Once the source code freeze is over for the specific release, the product needs to be built under the SCM environment. The building of the product makes the needed libraries and executables of the product. It is best practice to have nightly builds for the product. If build fails, the respective developer gets the e-mail from the system so that he can make the correction and checkin the source code into the version control system. Once build is successful, we can proceed to packaging.

7.2 Packaging

Packaging is needed for any product because the users should be able to install the product on their own in most of the cases. There are packaging tools such as InstallShield available in the market for software product packing purposes. Packaging tools provide facilities to package the patches as well.

7.3 Configuration Management

Configuration Management is important for any product environment. The product deliverables such as design documents and source code need to be maintained versions for making different releases. Commercially available SCM tools include Source Safe, CVS, Subversion, ClearCase and PVCS which can be used to maintain different versions of the files, checkin, checkout, branching and labeling of the products can be done.

8. Making Regular Releases

Based on the product roadmap, product management team can plan different releases of the product. Using SCM tools, Building and Packaging tools, IDEs, and with proper project management plans, design documents, help to make different releases of the product.

9. Maintaining the Product

Once product is released into the market, it enters the maintenance phase. The support team's job starts with that. It has to respond to the customer queries and update the product development team. Sometimes, the customer queries may result into a bug in the product. Then the maintenance team (if exists) or the development team has to fix the bug. Once certain number of bug fixes is over then the company may release a patch or incremental release of the product with additional features and bug fixes.

10. Different Types of Software Products

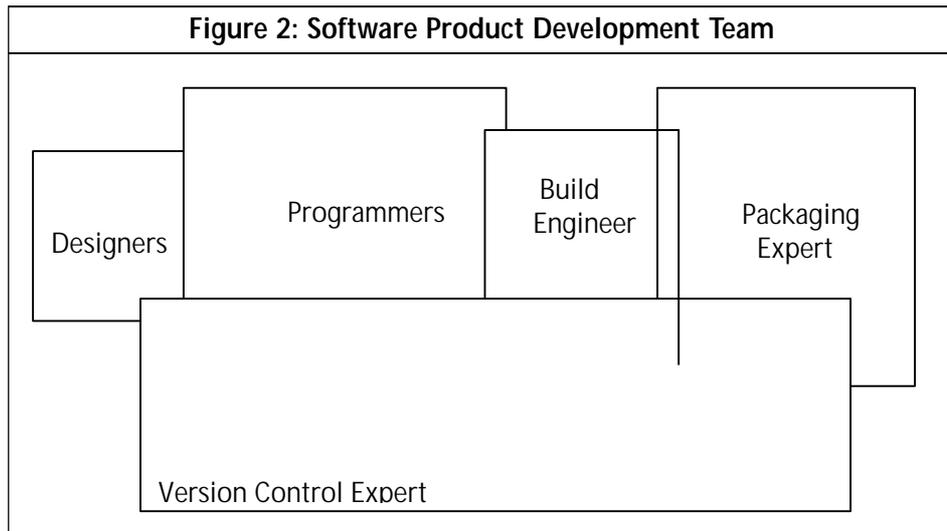
Basically the product development process is same for all types of software products whether it is an enterprise, web based, desktop, or stand-alone application. The editing, compiling, linking, building, debugging, packaging, and configuring vary from platform to platform based on the technologies we are using for the product development. IDE tools are different for different platforms such as Windows, Solaris, and HP-UX, etc. However any software product will have the phases of analysis, design, coding and testing as its product development steps.

11. Project Management

The project manager or the product manager responsible for the engineering side of the product has to manage the specific projects. His responsibilities include managing time, cost, scope, quality, human resources, communications, risks and procurement. The project manager needs to have the experience and skills in communication, decision-making, negotiation, leadership and technical skills. Project Management tools such as Microsoft Project, Primavera can be used to manage the time, cost and schedules. There are even tools for the project portfolio management available in the market. Now-a-days lightweight agile methodologies such as Scrum, Extreme Programming (XP), Adaptive Software Development (ASD), Crystal, and Feature Driven Development (FDD) are gaining support in the software industry. These software project management methodologies are useful for small teams.

12. Team Organization

The product development team consists of Designers, Programmers, Building and Packaging experts. Sometimes responsibilities of each individual may overlap with different roles. They are as shown in the Figure 2.



13. Conclusion

The teams which work on products are product development, testing, QA, support team, marketing team and strategy team. The different steps involved in the product engineering include requirements gathering, analysis, design, coding, testing, release and maintenance of the product. Different levels of design, testing and release management of the software product are explained which are also followed by the well-known companies in the world for their software product development.

(G P Sudhakar, Consulting Editor, Icfai Research Center, Hyderabad. The author can be reached at purna24@hotmail.com).

References

1. Bashar Nuseibeh and Steve Easterbrook (2000), "Requirements Engineering: A Roadmap", In A. C. W. Finkelstein (ed) "*The Future of Software Engineering*". (Companion volume to the proceedings of the 22nd International Conference on Software Engineering, ICSE'00). IEEE Computer Society Press, 2000.
2. Diomidis Spinellis & Clemens Szyperski (2004), "How Is Open Source Affecting Software Development", IEEE SOFTWARE, IEEE Computer Society, 2004.
3. Gerard J. Holzmann (2001), "Economics of Software Verification", Proceedings of Workshop on Program Analysis for Software Tools and Engineering, ACM, Snowbird, Utah, USA, June 2001.
4. Laura Wingerd & Christopher Seiwald (1998), "High-Level Best Practices in Software Configuration Management", Proceedings of the SCM-8 Symposium on System Configuration Management, 1998.
5. Mark Paulk, Charlie Weber, Suzanne Garcia, Mary Beth Chrissis, and Marilyn Bush (1993), "Key Practices of the Capability Maturity Model, Version 1.1", Software Engineering Institute, Carnegie Mellon University, 1993.
6. Rahul Chouda, "Software Product Management", available at <http://www.softwaremarketingresource.com/article10.html>
7. Rahul Sah (2008), "Software Testing Techniques", available at <http://www.ciol.com/Developer/Testing/Feature/Software-testing-techniques/17108102881/0/>
8. Roger S. Pressman (2001), "Software Engineering: A Practitioner's Approach", Fifth Edition, McGraw-Hill, 2001.
9. William Buxton (2003), "Performance by Design: The Role of Design in Software Product Development", Proceedings of the Second International Conference on Usage-Centered Design, Portsmouth, NH, USA, 26-29 October 2003, pp 1-15.

