Advanced Software Engineering Techniques

Course 2 – October, 19, 2015

Adrian Iftene
adiftene@info.uaic.ro
Content

- **Software Engineering Books**
  - GOF, GRASP, Swebok

- **Swebok**
  - Software configuration management
  - Software engineering management
  - Software engineering process
  - Software engineering tools and methods

- **Myths**
Software Engineering Books

- GOF (Gang–Of–Four) – Design Patterns: Elements of Reusable Object–Oriented Software, Erich Gamma, Richard Helm, Ralph Johnson, John Vissides

- GRASP – Applying UML and Patterns – An Introduction to Object–Oriented Analysis and Design and Iterative Development, Craig Larman
Swebok project

- **Software Engineering Body of Knowledge**

- Book’s authors Alain Abran, James W. Moore, 2004
Guide to the Software Engineering Body of Knowledge (SWEBOK)

Get the 2004 SWEBOK Guide
- HTML (free)
- PDF
- Book

SWEBOK Guide Set for V3 Refresh
Volunteers are gearing up to refresh the Guide to the Software Engineering Body of Knowledge—SWEBOK—intending to add new knowledge areas (KAs) and to revise others.

Developed concurrently, the SWEBOK Guide, the Software Engineering 2004 (SE2004) curriculum guide, and the Certified Software Development Professional (CSDP) certification each provided a characterization of the discipline of software engineering. Despite nearly independent development, the three instruments agreed to a remarkable extent. The primary purpose of the current revision of the SWEBOK Guide is to perfect the correspondence between the three items, notably by adding a KA on professional practices—a subject currently covered by the CSDP—and adding "foundation" KAs on related subjects that software engineers learn about during their undergraduate education—subjects currently covered by SE2004.

To achieve this alignment and to maintain the currency of the SWEBOK Guide, the IEEE Computer Society's Professional Practices Committee agreed in 2008 to the following changes:
- a new KA on Professional Practice—similar to material currently in the CSDP
The book is a Guide to the Software Engineering Body of Knowledge

In the book are defined 10 knowledge areas (KAs) in SE

- Software requirements
- Software design
- Software construction
- Software testing
- Software maintenance
- Software configuration management
- Software engineering management
- Software engineering process
- Software engineering tools and methods
- Software quality
Software configuration management

- SCM – the task of tracking and controlling changes in the software
  - revision control and
  - the establishment of baselines

- Q: "Somebody did something, how can one reproduce it?"
- A: comparing different results and of analyzing their differences
SCM Definition

- *SCM is a supporting* software life cycle process (IEEE12207.0–96) which benefits project management, development and maintenance activities, assurance activities, and the customers and users of the end product.

- The concepts of configuration management apply to all items to be controlled (both hardware and software).

- SCM is closely related to the software quality assurance (SQA) activity.
SCM Activities

- Management and planning of the SCM process, software configuration identification, software configuration control, software configuration status accounting, software configuration auditing, and software release management and delivery.

![SCM Activities Diagram]

Figure 1. SCM Activities
RC also known as version control, source control or software configuration management (SCM) is the management of changes to documents, programs, and other information stored as computer files.

Changes are usually identified by a number or letter code, termed the "revision number", "revision level", or simply "revision"
Each revision is associated with a **timestamp** and the **person** making the change.

For example, an initial set of files is "revision 1". When the first change is made, the resulting set is "revision 2", and so on.

Revisions can be compared, restored, and with some types of files, merged.
Stand-alone applications: Microsoft Word, OpenOffice.org Writer, KWord, Pages, Microsoft Excel, OpenOffice.org Calc, KSpread, Numbers

Content management systems: Drupal, Joomla, WordPress

In wiki software packages such as MediaWiki, DokuWiki, TWiki (offers the ability to revert a page to a previous revision. The aim is to correct mistakes, and defend public wikis against vandalism and spam)
**SCM – Vocabulary (1)**

- **Branch** – from that time forward, two copies of those files may develop at different speeds or in different ways independently.

- **Change/patch** – represents a specific modification to a document under version control.

- **Change list** – the set of changes made in a single commit.

- **Checkout** – creates a local working copy from the repository (we can specify a specific revision or obtain the latest).
Commit (checkin) – occurs when writing or merging a copy of the changes made to the working copy into the repository

Conflict – when different parties make changes to the same document, and the system is unable to reconcile the changes

Merge – two sets of changes are applied to a file or set of files
Tag – refers to an important snapshot in time, consistent across many files. Can be a user-friendly, meaningful name or revision number

Trunk – The unique line of development that is not a branch

Update – merges changes from repository into the local working copy

Working copy – is the local copy of files from a repository
Possible solution

- **Server:** ubuntu, apache2, SVN 
- [https://students.info.uaic.ro/infoiasi/svn/](https://students.info.uaic.ro/infoiasi/svn/)

- **Client:** Windows, TortoiseSVN 
Server

https://students.info.uaic.ro/infoiasi/svn/
Client
Eclipse (1)
Eclipse (2)
Eclipse (3)
The application of management activities—planning, coordinating, measuring, monitoring, controlling, and reporting—to ensure that the development and maintenance of software is systematic, disciplined, and quantified (IEEE610.12–90)
SE management – Aspects

- The perception of clients is such that there is often a lack of appreciation for the complexity inherent in software engineering.
- It is almost inevitable to generate the need for new or changed client requirements.
- Software is built in an iterative process rather than a sequence of closed tasks.
- Software engineering necessarily incorporates aspects of creativity and discipline—maintaining a balance between the two is often difficult.
- The degree of novelty and complexity of software is often extremely high.
- There is a rapid rate of change in the underlying technology.
Figure 1 Breakdown of topics for the Software Engineering Management KA
Can be examined on two levels

The first level encompasses the technical and managerial activities within the software life cycle processes

The second is the meta-level, which is concerned with the definition, implementation, assessment, measurement, management, change, and improvement of the software life cycle processes themselves
Figure 1 Breakdown of topics for the Software Engineering Process KA
Software engineering tools and methods (Swebok)

- **Tools** that are intended to assist the software life cycle processes, and to allow repetitive, well-defined actions to be automated, reducing the cognitive load.

- They are intended to make software engineering more systematic.

- **Methods** impose structure on the software engineering activity with the goal of making the activity systematic and successful.

- Methods provide a notation and vocabulary, procedures for performing identifiable tasks, and guidelines for checking both the process and the product.
Swebok – Related Disciplines

- Computer engineering
- Computer science
- Management
- Mathematics
- Quality management
- Software ergonomics (Cognitive ergonomics)
- Systems engineering
Computing Curricula 2001 project (CC2001) states that “computer engineering embodies the science and technology of design, construction, implementation and maintenance of software and hardware components of modern computing systems and computer-controlled equipment.”

KAs for computer engineering:
- Algorithms and Complexity
- Computer Architecture and Organization
- Computer Systems Engineering
- Circuits and Systems
- Digital Logic
- Discrete Structures
- Digital Signal Processing
- Distributed Systems
- Electronics
- Embedded Systems
- Human–Computer Interaction
- Information Management
- Intelligent Systems
- Computer Networks
- Operating Systems
- Programming Fundamentals
- Probability and Statistics
- Social and Professional Issues
- Software Engineering
- Test and Verification
- VLSI/ASIC Design
ISO Technical Committee 159 on Ergonomics as follows: “Ergonomics or (human factors) is the scientific discipline concerned with the understanding of the interactions among human and other elements of a system, and the profession that applies theory, principles, data and methods to design in order to optimize human well-being and overall system performance.”

KAs:

- Cognition
- Machine Learning and Grammar Induction
- Formal Methods in Cognitive Science: Language
- Formal Methods in Cognitive Science: Reasoning
- Information Extraction from Speech and Text
- Lexical Processing
- Computational Language Acquisition
- Human–Machine Fit and Adaptation
- Human Characteristics
- Computer System and Interface Architecture
- Dialogue Architecture
- Development Process
Myths

- **Clients**
  - A general description of the objectives is sufficient to begin writing program
  - Requirements are constantly changing, but the software is flexible and can easily adapt

- **Developers**
  - Once the program is written and it is functional, our role has ended
  - Until the program doesn’t work, we can not assess the quality
  - The only good product is the functional program
  - Software Engineering will create voluminous and unnecessary documentation and will cause delays
Links

Bibliografie


- Erich Gamma, Richard Helm, Ralph Johnson, John Vissides: Design Patterns, Elements of Reusable Object-Oriented Software, Addison Wesley, 1998