Software Engineering Essentials

Introduction

Mohamad Halabi
@mohamadhalabi

pluralsight
hardcore dev and IT training

A Discipline in Demand

- **Software Engineering is a highly demanded discipline and Software Engineers are much sought after**
- **Mashable report: Software Engineer is the top tech job of 2014**
  - http://mashable.com/2014/01/06/tech-jobs-2014/
- **Business Insider report: Software Engineers are highly paid pros in the biggest companies**
  - http://www.businessinsider.com/software-salaries-at-big-companies-2014-9

…Yet a Misunderstood Discipline

- What is software engineering and why is it important?
- What are the building blocks of software engineering?
- What are the processes and methods that differentiate it as a discipline?

What Is Software Engineering?

…the application of a systematic, disciplined, and quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software…

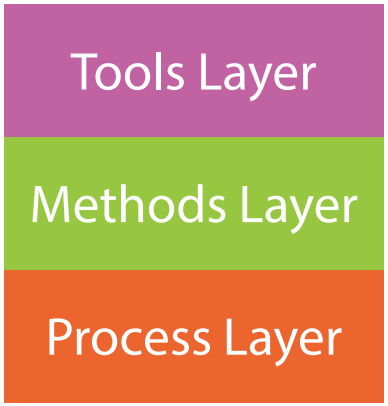
— IEEE

Applied through entire lifecycle: specifications → maintenance

Why Software Engineering?

- **The alternative is ad-hoc or disordered approach**
- **Engineering means:**
  - **Predictability** and **quantifiable** results
  - Application of **theories**, **methodologies**, **frameworks**, and **tools**
  - Result is **high-quality** software created in **cost-effective** manner

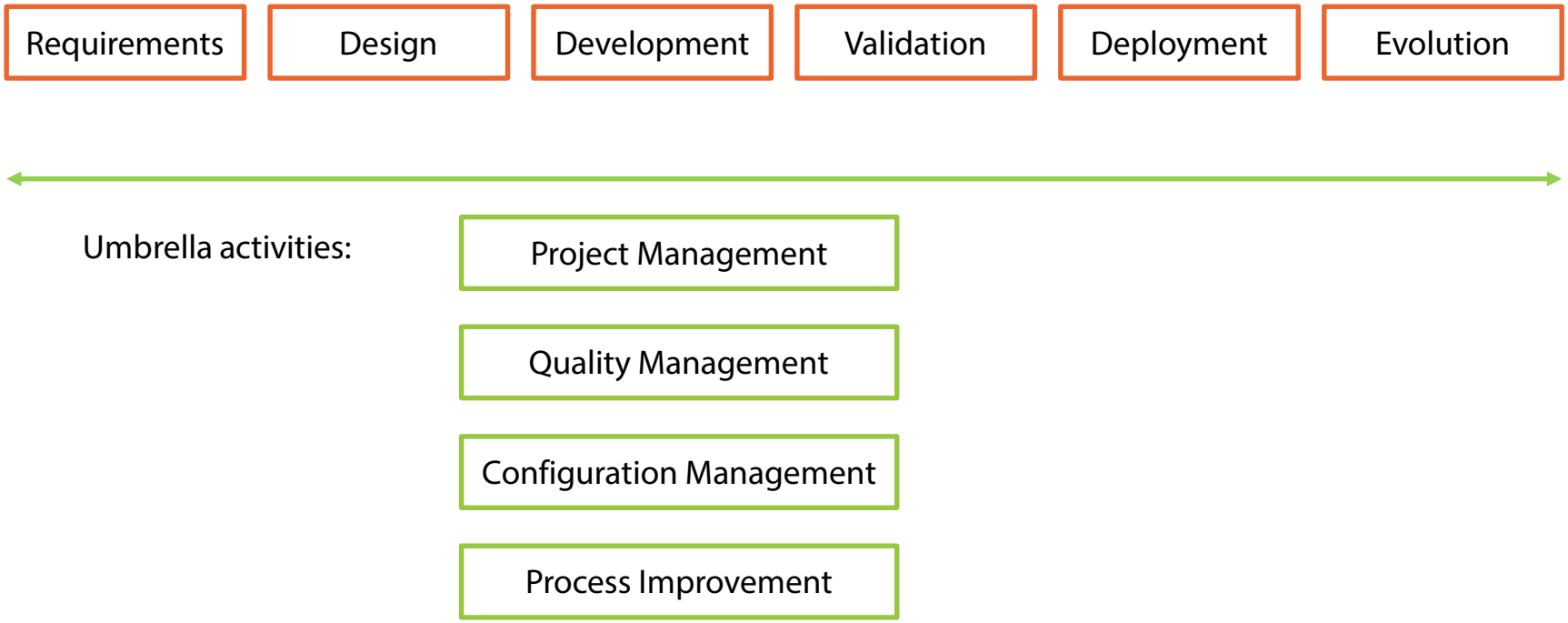
Software Engineering Layers



- **Process Layer**
  - Framework and order of activities
  - How Requirements, Design, Construction, and Testing are performed?
- **Methods Layer**
  - Proven techniques to perform certain activities
  - Ex: methods for requirements analysis/modeling, design and design modeling, and testing, etc…
- **Tools Layer**
  - Provides automation support
  - Aids in the systematic application of software engineering

Software-Creation Activities

Generic set of software lifecycle activities:

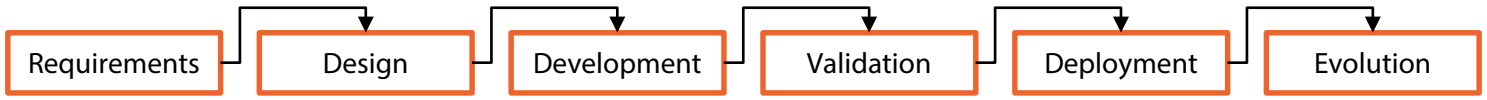


Software Process

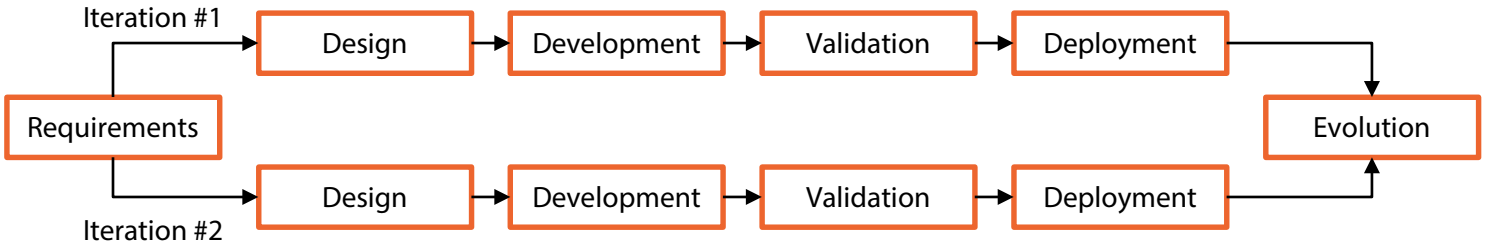
- **Also called Software Development Lifecycle (SDLC)**
- **Defines:**
  - Tasks inside the Software Engineering activities
  - Order and detail of these tasks and activities
  - Flow of activities (ex: iterative, linear, etc…)
  - Type and detail of artifacts

Software Process

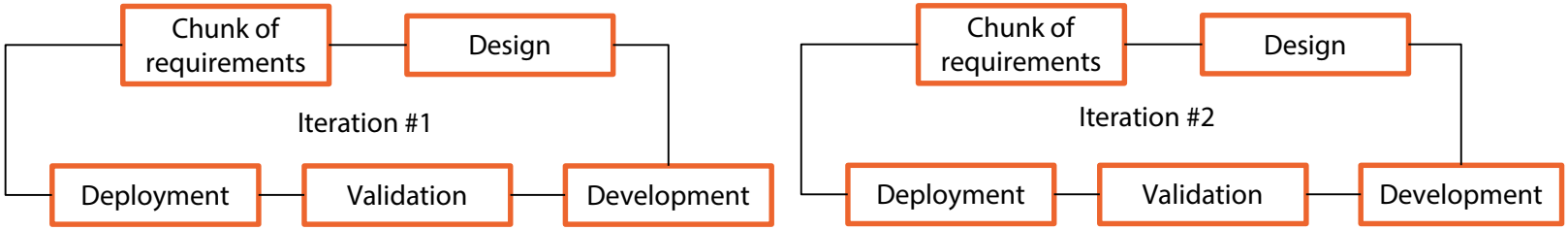
Linear model?



Iterative model?



Agile model?



What about the artifacts?

Software Process Models

- **Traditional (classic) models**
  - Waterfall (linear)
- **Iterative and incremental models**
  - Prototyping
  - Spiral
  - Agile
  - Unified Model
- **Specialized models (for particular approaches)**
  - Component-based development
  - Formal methods
  - Aspect-Oriented development

Which Model to Use?

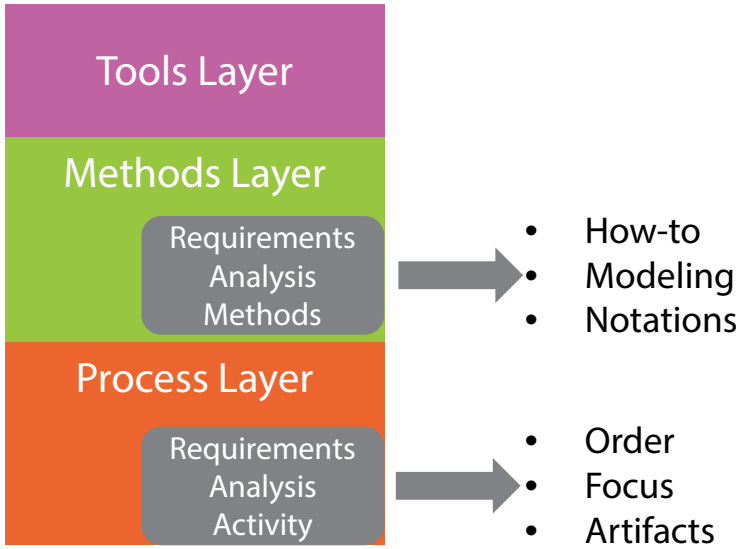
- **Decided by various factors, including project type and organization maturity**
  - Ex: Airplane navigation system requires rigorous and detailed specification and design
  - Ex: Collaboration portal does not need such rigidity
- **Typically, the project manager with the lead software engineer decide on the process model**

Alert: A Software Process Spans Entire Lifecycle

- **Software professionals often make common mistakes:**
  - They do not consider maintenance (evolution) as part of the software process
  - They consider Requirements → Deployment but neglect operations
- **This is wrong! Maintenance is part of the software lifecycle (until the system is retired)**
- **Keep in mind: pre-deployment requests (i.e. changes) also need analysis, design, development, testing, and deployment**
  - Often as part of a change management process (ex: ITIL) rather than a project charter

Software Engineering Methods

- **Practices with proven techniques to perform certain activities in an organized and systematic approach**



- **Similarly, there are methods for design, testing, etc…**
- **Some of the most known Software Engineering methods:**
  - Structured Analysis and Design
  - Object-Oriented Analysis and Design
  - Formal Methods

Software Engineering Tools

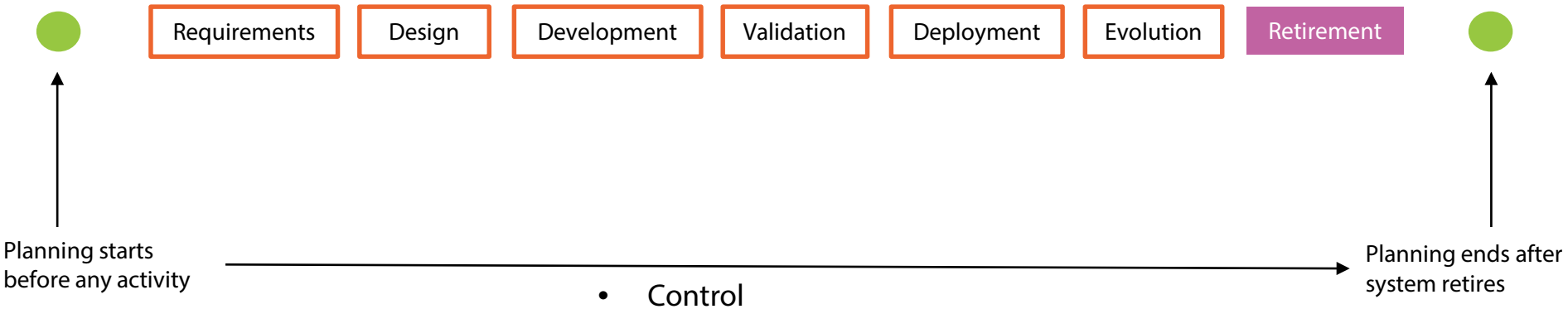
- **Tools assist software processes by automating actions**
- **Examples**
  - Requirement Management: Enterprise Architect (Sparx), IBM Rational (IBM)
  - Development: Visual Studio (Microsoft), Eclipse (Open Source)
  - Testing: Team Foundation Server (Microsoft), HP Quality Center (HP)
  - Software Configuration: Team Foundation Server (Microsoft)
  - Project Management: Enterprise Project Management (Microsoft)
  - Process Management and Modeling: Team Foundation Server

The Role of Software Engineer

- **Software Engineer as a role is less formally defined than the discipline**
- **Software Engineer is replacing the title Software Developer**
  - Seems to indicate more knowledge/responsibilities. So more sophisticated!
- **However, Software Engineering is not only about development**
  - So is a designer or tester also a Software Engineer?
- **In general, anyone who efficiently applies the engineering discipline to the analysis, design, development, testing, and operation is a qualified Software Engineer**

Software Engineering and Project Management

- **We already established that:**
  - Project Management is an umbrella activity
  - Software Engineering is applied throughout the software lifecycle



- Control
- Communication
- Cost analysis
- Work breakdown structure
- Task scheduling
- Risk analysis

SWEBOK

- **IEEE Computer Society publishes the Software Engineering Body of Knowledge (SWEBOK) as an international standard**
- **SWEBOK (v3) promotes consistent view and specifies scope of Software Engineering**
- **Software Engineering activities are organized into 15 Knowledge Areas**

1. Software Requirements
2. Software Design
3. Software Construction
4. Software Testing
5. Software Maintenance
6. Software Configuration Management
7. Software Engineering Management
8. Software Engineering Process
9. Software Engineering Models and Methods
10. Software Quality
11. Software Engineering Professional Practice
12. Software Engineering Economics
13. Computing Foundations
14. Mathematical Foundations
15. Engineering Foundations

Content

- **Building blocks of Software Engineering: Process, Methods, and Tools**
- **Module 2: tour around process models**
  - Activities are explained independent of the process model applied
  - Activities are standardized. Process models set the ceremony level
- **Module 3: requirements engineering process**
- **Module 4: requirements modeling Structured Analysis method**
- **Module 5: requirements modeling Object-Oriented Analysis method**
- **Module 6: essentials of software design process**
- **Module 7: design method**
- **Module 8: construction process**
- **Module 9: testing process and methods**

Summary

- ***"…the application of a systematic, disciplined, and quantifiable approach to the development, operation, and maintenance of software…"*** **-IEEE**
  - So Software Engineering is applied to the entire software lifecycle
- **Three layers are the building blocks of Software Engineering:**
  - Process Layer:
    - Activity tasks (i.e. requirements, design, development, testing, deployment, evolution)
    - Order and detail of these activities
    - Flow (ex: iterative or linear)
    - Type and detail of artifacts
    - There are different process models (ex: agile, waterfall, unified process)
  - Methods Layer: Practices with proven techniques to perform certain activities
  - Tools Layer: automation support to perform activities