RSET
RAJAGIRI SCHOOL OF ENGINEERING & TECHNOLOGY

Lecture - 2

Topics are:

Legacy Software

A generic view of software – A layered technology

A process framework

RSET
RAJAGIRI SCHOOL OF ENGINEERING & TECHNOLOGY

Legacy Software

The word legacy denoting or relating to software or hardware that has been superseded(-> refers a place) but is difficult to replace because of its wide use.

Dayani-Fard describe the legacy software in the following way:

Legacy Software Systems were developed decades ago and have been continually modified to meet changes in business requirements and computing platforms. The proliferation (rapid increase in the number or amount of something) of such systems is causing headaches for large organizations who find them costly to maintain and risky to evolve.

Characteristics of Legacy Software

- Longevity (Lifetime) : Very high
- Business criticality : High
- Poor quality
- Convoluted code
- Poor or nonexistent documents, test cases.

However, as time passes legacy systems often evolve for one or more following reasons.

Organizations can have compelling reasons for keeping a legacy system, such as:

- The system works satisfactorily, and the owner sees no reason to change it.
- The costs of redesigning or replacing the system are prohibitive because it is large and complex.
- Retraining on a new system would be costly in lost time and money, compared to the anticipated appreciable benefits of replacing it (which may be zero).
- The system requires near-constant availability, so it cannot be taken out of service, and the cost of designing a new system with a similar availability level is high. Examples include systems to handle customers' accounts in banks, computer reservations systems, air traffic control etc.
- The way that the system works is not well understood. Such a situation can occur when the designers of the system have left the organization, and the system has either not been fully documented or documentation has been lost.

Changes needed in legacy software

1. The software must be adopted to meet the needs of new computing environments or technology.
2. The software must be enhanced to implement new business requirements.
3. The software must be extended to make it interoperable with more modern systems or databases.
4. The software must be re-architected to make it viable within a network environment.

A Generic view of Software

Meaning of generic: characteristic of or relating to a class or group of things; not specific.

Aims to design a high-quality software.

Software process as a framework(an essential supporting structure) for the tasks that are required to build high-quality software. It requires communication, planning, modeling, construction, and deployment.

Software engineering is always a layered technology.

Software Engineering – A Layered Technology

Does any technology behind a software? Yes.

Listen the definition:

Software engineering is the establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines.

IEEE has another definition:

Software Engineering: (1) The application of a systematic, disciplined, quantifiable approach to the development, operation and maintenance of software; that is, the application of engineering to software. (2) The study of approaches as in (1).

RSET
RAJAGIRI SCHOOL OF ENGINEERING & TECHNOLOGY

Layered Technology – Different Layers in a software

RSET
RAJAGIRI SCHOOL OF ENGINEERING & TECHNOLOGY

Bedrock – Bottom most Layer "Quality"

Bedrock (Bottom most layer): Most engineering approaches (including software engineering) must rest on an organizational commitment to quality. The bedrock that supports software engineering is a quality focus layer.

Quality: a product should meet its specification. This is problematical for software systems. There is a tension between customer quality requirements (efficiency, reliability, etc.), developer quality requirements (maintainability, reusability, etc.), users (usability, efficiency, etc.), and etc.

Note: Some quality requirements are difficult to specify in an unambiguous way. Ex:- Software specifications are usually incomplete and often inconsistent.

Layer-2 “Process”

The foundation for software engineering is the process layer.

It is a glue that holds the technology together and enables rational and timely development of computer software.

The work products are produced, milestones are established, quality is ensured, and changes are properly managed.

Layer 3 “Methods”

Software engineering methods provides the technical “how-to‘s” for building software.

Methods encompass a broad array of tasks that include the requirements analysis, design, program construction, testing, and support.

RSET
RAJAGIRI SCHOOL OF ENGINEERING & TECHNOLOGY

Top most Layer “Tools”

Software engineering tools provide automated or semi-automated supports for the process and the methods. When the tools are integrated so that the information created by one tool can be used by another, a system for the support of software development, called computer-aided software engineering (CASE).

CASE combine software, hardware, and software engineering database.

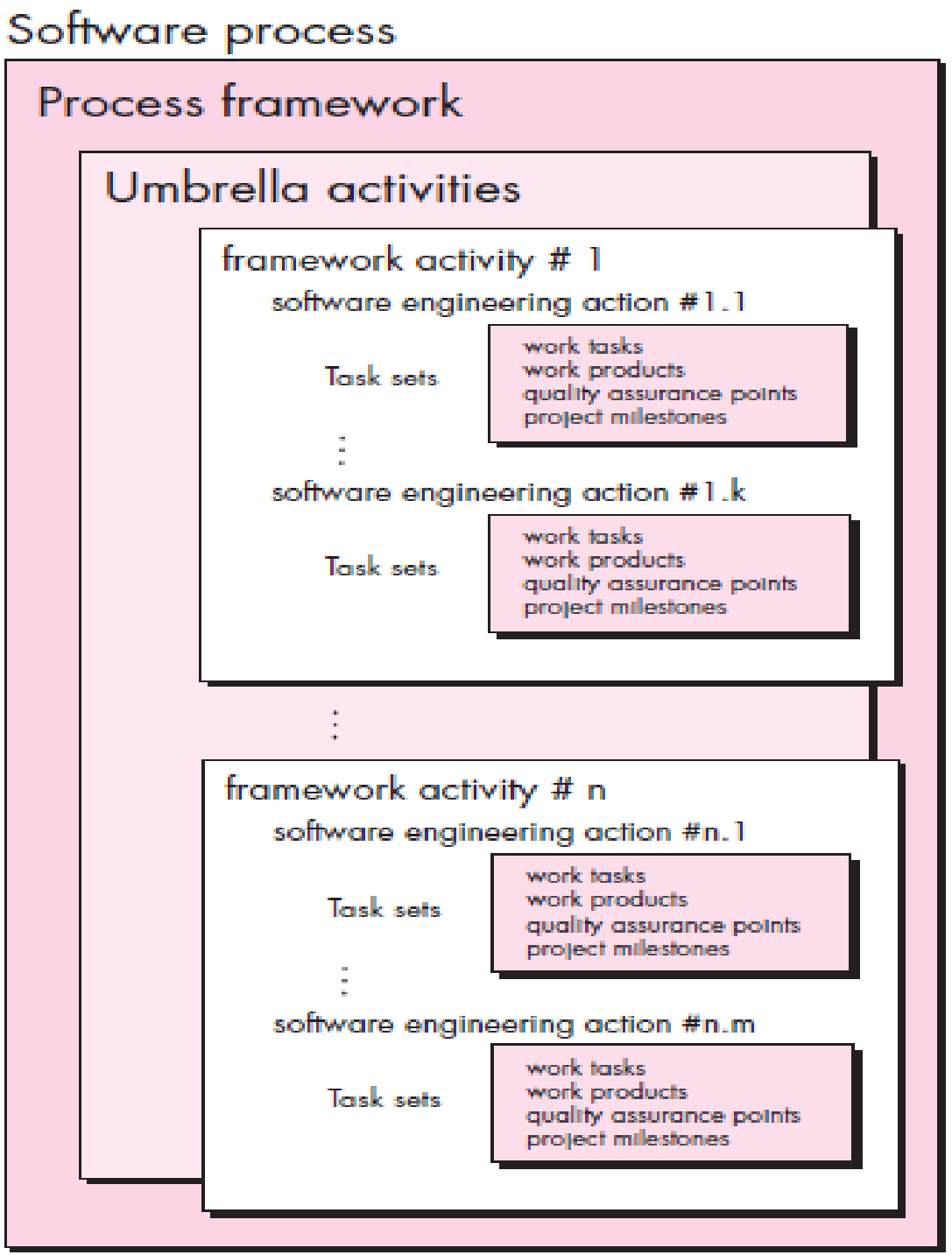
RSET
RAJAGIRI SCHOOL OF ENGINEERING & TECHNOLOGY

A Process Framework

Definition: A process framework establishes the foundation of a complete software process **by identifying a small number of framework activities** that are applicable to all software projects, regardless of their size and complexity.

In addition, a **process framework encompasses a set of umbrella activities** that are applicable across the entire software process.

Each framework activity is populated by a set of **software engineering actions**.


Software process
Process framework
Umbrella activities
framework activity # 1
software engineering action #1.1
Task sets
work tasks
work products
quality assurance points
project milestones
software engineering action #1.k
Task sets
work tasks
work products
quality assurance points
project milestones
framework activity # n
software engineering action #n.1
Task sets
work tasks
work products
quality assurance points
project milestones
software engineering action #n.m
Task sets
work tasks
work products
quality assurance points
project milestones

Five Generic Framework Activities (Micro-activity)

1. **Communication** – heavy communication and collaboration with customers.
2. **Planning** – technical tasks to be completed, the risks that are likely, the resources that will be required, the work products to be produced, and a work schedule.
3. **Modeling** – creation of new models or prototypes to make both developer and customer to understand the system.
4. **Construction** – Combines code generation and software testing.
5. **Deployment** – The customer receives the product at this stage. A customer needs to evaluate it and provides feedback to software developers.

RSET
RAJAGIRI SCHOOL OF ENGINEERING & TECHNOLOGY

Eight Umbrella Activities

1. Software project tracking and control – allows the software team to assess progress against the project plan. They take necessary action to maintain schedule.
2. Risk management – assesses risks that may effect the outcome
3. Software quality assurance – defines and conducts the activities required to ensure software quality.
4. Formal technical reviews – assesses software work products to uncover errors before they are propagated to next activity.
5. Measurement – defines and collects process, project and product measures that assist the team in delivering software that meets customers' requirement.
6. Software configuration management – manages the effects of change throughout the software process.
7. Reusability management – defines criteria for work product reuse and establishes mechanism to achieve reusable components.
8. Work product preparation and production – encompasses the activities required to create work products such as models, documents, logs, forms and lists.