UECS3383 Software Quality Assurance

Lecture 04 – Risk Management

Why Development and Quality Plan?

- It is often the case in the software industry that project managers (PMs) come under serious time pressures from the very first day
- PMs rely on development plan and quality plan because PMs know that the proposal and its estimates, including the timetable, staff requirements, list of project documents, scheduled design reviews, and list of development risks have all been thoroughly reviewed by the contract review team.

Criteria of Development Plan and Quality Plan

- Are based on proposal that have been re-examined and thoroughly updated
- Are more comprehensive than the approved proposal, especially with respect to schedules, resource estimates and development risk evaluations
- Include additional subjects, absent from the approved proposal
- Were prepared at the beginning of the project to sound alert regarding scheduling difficulties, potential staff shortages, paucity of development facilities, problems with meeting contractual milestones, modified development risks, etc.

Objectives of Development and Quality Plans

- Planning is a **process** meant to prepare adequate foundations for successful and timely completion of the project. The planning process includes:
  - Scheduling development activities and estimating the required manpower resources and budget
  - Recruiting team members and allocating development resources
  - Resolving development risks
  - Implementing required SQA activities
  - Providing management with data needed for project control

The Elements of a Development Plan

1. Project products, specifying “deliverables”
2. Project interfaces
3. Project’s methodology and development tools
4. Software development standards and procedures
5. Map of the development process
6. Project milestones
7. Project staff organization and coordination with external participants
8. Required development facilities
9. Development risks and risk management actions
10. Control methods
11. Project cost estimates

The Elements of a Development Plan

- **Project products, specifying "deliverables" such as:**
  - Design documents specifying dates of completion, indicating those items to be delivered to the customer ("deliverables")
  - Software products (specifying completion date and installation site)
  - Training tasks (specifying dates, participants and sites)
- **Project interfaces includes:**
  - Interfaces with existing software packages (software interface)
  - Interfaces with other software and/or hardware development teams that are working on the same system or project (i.e. cooperation and coordination links)
  - Interfaces with existing hardware (hardware interface)

The Elements of a Development Plan

- **Project's methodology and development tools** to be applied at each phase of the project
- **Software development standards and procedures** – a list should be prepared of the software development standards and procedures to be applied in the project
- **Map of the development process** involves providing detailed definitions of each of the project phases. These descriptions include definitions of inputs and outputs, and the specific activities planned. **Activity descriptions include**:
  - An estimate of the activity's duration. These estimates are highly dependent on the experience gained in previous projects
  - The logical sequence in which each activity is to be performed, including a description of each activity's dependence on previously completed activities
  - The type of professional resources required and estimates of how much of these resources are necessary for each activity
  - Several methods are available for scheduling and graphically presenting the development process (e.g. Gantt chart)

The Elements of a Development Plan

- **Project milestones**
  - For each milestones, its completion time and project products (documents and code) are to be defined
- **Project staff organization and coordination with external participants**. **The organization plan comprises**:
  - Organization structure
  - Professional requirements
  - Number of team members required for each period of time
  - Names of team leaders and team members.
  - Coordination with external participations

The Elements of a Development Plan

- **Required development facilities** – required development facilities include hardware, software and hardware development tools, office space and other items. For each facility, the period required for its use should be indicated on the timetable
- **Control methods** – in order to control project implementation, the PMs and the department management apply a series of monitoring practices when preparing progress reports and coordinating meetings.
- **Project cost estimates** are based on proposal costs estimates, followed by a thorough review of heir continued relevance based on updated human resource estimates, contracts negotiated with subcontractors and suppliers, and so forth.

The Elements of a Development Plan

- **Development risks and risk management actions**
  - To understand their pervasiveness, and how they can be controlled, the team should first define the concept
  - A development risk is "a state or property of a development task or environment, which, if ignored, will increase the likelihood of project failure
  - Typical development risks are:
    - Technological gaps
    - Staff shortages
    - Interdependence of organizational elements

Important Notes to the Development Plan

- When evaluating the suitability of propose project methodology and development tools, one should take into account the professional experience of the staff, including the subcontractors' personnel, even if temporary
- SQA activities, such as **design review** and **software tests**, should be included among the scheduled project activities. The same applies to the design and code **correction activities**. Failing to schedule these activities can cause unanticipated delays in the initiation of subsequent activities
- The long-term availability of project staff should be carefully examined. Lags in completing former assignments may result in delays in joining the project team, which increases the risk of failing to meet project milestones. In additional, staff "evaporation" caused by resignations and/ or promotions, phenomena that are particularly frequent in the software industry, can cause staff shortages. Therefore, estimates of staff availability should be examined periodically to avoid "surprises". Early warning of unforeseen staff shortages make it easier to resolve the problem

The Elements of a Software Quality Plan

1. List of quality goals
2. Review activities
3. Software tests
4. Acceptance tests for software externally developed
5. Configuration management plans: tools, procedures and data for version releases

The Elements of a Software Quality Plan

- **List of quality goals**
  - "Quality Goals" refers to the developed software system's substantive quality requirements
  - Prefer qualitative measures because they provide the developer with more objective assessments of software performance during the development process and system testing
  - However, one type of goal is not totally equivalent to the other. There is a possibility replacement of qualitative with quantitative measures or vice versa
  - Quality goals should reflect the major acceptance criteria indicated in the customer's requirement document.
  - Quality serves as measures of the successful achievement of the customer's quality requirements

The Elements of a Software Quality Plan

- **List of quality goals** – an example illustration

HDS Qualitative Requirements	Related Quantitative Quality Goals
The HDS should be user friendly	A new helpdesk operator should be able to learn the details of the HDS following a course lasting less than 8 hours, and to master operation of the HDS in less than 5 working days
The HDS should be very reliable	HDS availability should exceed 99.5% (HDS downtime should not exceed 30 minutes per week)
The HDS should operate continuously	The system's recovery time should not exceed 10 minutes in 99% of cases of HDS failure
The HDS should be highly efficient	An HDS operator should be able to handle at least 100 customers calls per 8-hour shift
The HDS should provide high quality service to the applying customers	Waiting time for an operator response should not exceed 30 seconds in 99% of the calls. Achievement of this goals depends on the combination of software features and number of workstations installed and operated

The Elements of a Software Quality Plan

- **Review activities**
  - Quality plan should provide a complete listing of all planned review activities: design reviews (DRs), design inspections, code inspections, and so on, with the following determined for each activity:
    - The scope of the review activity
    - The type of the review activity
    - The schedule of review activities (as defined by its priority and the succeeding activities of the project process)
    - The specific procedures to be applied
    - Who is responsible for carrying out the review activity?
- **Configuration management plans:** tools, procedures and data for version releases

The Elements of a Software Quality Plan

- **Software tests**
  - Quality plan should provide a complete list of planned software tests, with the following designated for each test:
    - The unit, integration or the complete system to be tested
    - The type of testing activities to be carried out, including specification of computerized software tests to be applied
    - The planned test schedule (as defined by its priority and the succeeding activities of the project process)
    - The specific procedures to be applied
    - Who is responsible for carrying out the test

The Elements of a Software Quality Plan

- **Acceptance tests for software externally developed**
  - A complete list of acceptance test planned for externally developed software should be provided within the quality plan
  - Items to be included are:
    - Purchased software
    - Software developed by subcontractors
    - Customer-supplied software
  - The acceptance tests for externally developed software should parallel those used for internally developed software tests

Software Quality Plan – Format and Approval

- The quality plan may be prepared as part of the development plan or as an independent document
- In some cases, the plan is divided into several documents by items category, such as DR plan, testing plan, and plan for externally developed software acceptance tests
- Review and approval of the quality plan should be conducted according to the organization's standard procedures for such plans

Benefits of Preparing Development and Quality Plans

- It is recommended that internal projects, undertaken on behalf of other departments and for development of software packages geared toward the market, be treated as "regular projects". This implies that full-scale development and quality plans are to be prepared. Benefits includes:
  - **The development department** will avoid losses incurred by unrealistic timetables and budgets, as well as the consequent damage to other projects and to the firm's reputation
  - **The internal "customer"** will enjoy reduced risk of late completion and budget overruns in addition to and by improved project control and coordination's with the developer
  - **The firm** will enjoy reduced risk of its software product's late entry into the market, reduced risk of a decline in its reputation resulting from late supply, and reduced risk of budget overruns

Classes of Software Development Risks

1. Scheduling and timing risks
2. System functionality risks
3. Subcontracting risks
4. Requirement management risks
5. Resource usage and performance risks
6. Personnel management risks

Top 10 Software Risk Items (SRIs)

(Boehm and Ross's)

1. Developing wrong software functions
2. Unrealistic schedules and budgets
3. Developing wrong user interface
4. Gold plating (additional of unnecessary features due to professional interests, pride or user demand)
5. Continuing stream of requirement changes
6. Shortfalls in externally furnished components
7. Shortfalls in externally performed tasks
8. Personnel shortfalls
9. Real-time performance shortfalls
10. Straining computer science capabilities

Risk Management Activities and Measures

- Risk Management Action (RMAs) is to prevent software risks to achieve early identification of software risk items and to resolve them
- The RMAs can be grouped into the following classes:
  - **Internal risk management** actions, applied within the software developing organization
  - **Subcontracting risk management** actions, dealing with the relationship between the software developer ad his subcontractors and suppliers
  - **Customer risk management** actions, dealing with the relationship between the software developer ad the customer
- In planning RMAs, one should aware that:
  - Some RMAs can prevent, identify or resolve SRIs of various types
  - Some SRIs can be treated by several RMAs
  - The efficiency of an RMA varies significantly with different projects and in different environments

Software Risk Management Process

Activities of the Risk Management Process

1. Risk Identification
2. Risk Evaluation
3. Planning of Risk Management Actions (RMAs)
4. Implementation of RMAs
5. Monitoring of RMAs

Activities of the Risk Management Process

- **Risk Identification**
  - The main tool supporting identification of SRIs is those checklists that specify the team, project and customer situations that are likely to bring about software risks
  - Identification of software risk items should begin with the actual start of the project (pre-project stage) and be repeated periodically throughout the project until its completion
- **Risk Evaluation** concerned mainly with:
  - Estimating of the probability that a software risk will materialize if no RMA is taken – i.e. **Prob(mat)**
  - Estimating damages in case an SRI does materialize – i.e. **Est(dam)**
  - SRI showing high Prob(mat) and Est(dam) is of high priority
  - SRI showing low Prob(mat) and Est(dam) is of low priority
  - One way to prioritize SRIs is by calculating their expected damages, termed "**risk exposure**" – **Exp(risk)**
  - **Exp(risk) = Est(dam) x Prob(mat)**

Activities of the Risk Management Process

- **Planning of Risk Management Actions (RMAs)**
  - RMAs include a range of internal, subcontractor and customer actions
  - The RMAs planning team should considered:
    - The priority assigned to the SRI
    - The expected results of planned RMA (complete or partial resolution)
    - The cost and organizational efforts required for implementation of the RMA
- **Implementation of RMAs**
  - Staff members required to be assigned personally responsible for EACH RMA and its implementation schedule
- **Monitoring of RMAs**
  - Required systematic and periodic activities to monitor implementation of the RMA plan
  - The aim of the monitoring activities is to:
    - Determine the efficiency of the RMAs
    - Update the risk evaluation by considering newly identified SRIs

Recommended Risk Management Actions and Their Contributions

Software RMA	Contribution of the RMA		
Internal RMA	Prevention	Early Identification of SRI	Resolution of SRI
Application of detailed and thorough analysis to requirements and estimated schedule and costs	✓		
Efficient project organization, adequate staff and team size	✓		
Personnel training	✓		
Arranging for and training replacements to take over in case of turnover and unanticipated workload	✓		
Arranging for user participation in the development process	✓		
Applying efficient change control (change requests screening)	✓		
Applying intensive software quality assurance measures such as inspections, design reviews and benchmarking	✓		
Project progress control of internally and externally performed activities including periodic checking for timely availability of firm professionals currently occupied with other projects		✓	
Arranging for participation of professional staff members having knowledge and experience with SRIs			✓

Recommended Risk Management Actions and Their Contributions

Software RMA	Contribution of the RMA		
Internal RMA	Prevention	Early Identification of SRI	Resolution of SRI
Scheduling SRI-related activities as early as possible to provide leeway in case of difficulties			✓
Prototyping SRI-related modules or project applications			✓
Preparing scenario for complicated SRI-related modules or project applications			✓
Simulating FRI-related modules or project application			✓
Subcontracting RMA			
Preparing comprehensive and thorough contracts with subcontractors and suppliers, including contract reviews	✓		
Participating in internal progress control and software quality assurance activities of subcontractors to be incorporated in the contract		✓	
Arranging for “loans” of professional with specialized knowledge and experience if the need arises			✓
Hiring consultants to support the team in the absence of sufficient know-how and experience			✓

Recommended Risk Management Actions and Their Contributions

Software RMA	Contribution of the RMA		
Customer RMA	**Prevention**	**Early Identification of SRI**	**Resolution of SRI**
Formulating comprehensive and thorough contracts with customers, including contract reviews	✓		
Negotiating with the customer to change requirements regarding risky parts of the project			✓
Negotiating with the customer to change schedules regarding risky parts of the project			✓

Software Safety and Hazard Analysis

- Software is frequently overlooked during system safety and hazard analyses, but this is unacceptable when the software is in control of a potentially hazardous operation. In such cases, software safety and hazard analysis should be extended to fully cover the software.

- Software safety ensures that the software in the computer system will execute within the application system without resulting in unacceptable risk.

- Hazard analysis is a method of identifying portions of a system which have the potential for unacceptable hazards; the purpose is to (1) encourage design changes which will reduce or eliminate hazards, or (2) carry out special analyses and tests which can provide increased confidence in especially vulnerable portions of the system.

Terminology

- The term **safety** is used to mean the extent to which a system is free from system hazard
- **Hazard analysis** is the process of identifying and evaluating the hazards of a system, and then either eliminating the hazard or reducing its risk to an acceptable level
- **Software hazard analysis** ". . . eliminates or controls software hazards and hazards related to interfaces between the software and the system (including hardware and human components). It includes analyzing the requirements, design, code, user interfaces and changes.

Software Safety and Hazard Analysis

- Software hazard analysis should be performed within the context of the overall system design for both those attributes of the system design that contribute to the system's ability to perform the assigned tasks that are derived from the software's safety mission.

- Those performing the software hazard analysis must understand the role of the software in the performance of the system safety functions and also in the performance of the system control and monitoring functions, and the effect of the software acting within the system with respect to its potential impact on the accomplishment of the plant's safety mission. This understanding is obtained from the system safety analysis; in particular, the system's hazard analysis

Software Safety and Hazard Analysis – Software Design

- The objectives of any hazard analysis program are to identify and correct deficiencies and to provide information on the necessary safeguards.

- At least four types of appropriate actions depending on the circumstances:
  - The system design may be changed to eliminate identified hazards which are affected by software or are not adequately handled by software, or to reduce the hazards to acceptable levels, or to adjust the system architecture so that identified hazards are compensated by defense-in- depth.
  - The software design may be changed to eliminate identified hazards, or reduce them to acceptable levels.
  - The quality of the software may be improved sufficiently to reduce the probability of a hazard to an acceptable level.
  - The application system may be rejected if it is considered too hazardous.

General Approach to Software Safety and Hazard Analysis

- Software hazard analysis should be a defined aspect of the software life cycle. No specific life cycle is endorsed here (see Lawrence (1993) for a discussion of life cycles).

- The general approach shows the technical development activities (requirements, architecture, design, code), the V&V activities, and the hazard analysis activities. Results of the various software hazard analyses are used, as appropriate, to change the protection system design, change the software architecture or design, and to identify portions of the software which require increased attention to quality.

Software Safety and Hazard Analysis Procedures

- The steps are meant to help organize the process. Variations in the process, as well as overlap in time among the steps, is to be expected.
  - Identify the hazards for which software is in any way responsible. This identification includes an estimate of the risk associated with each hazard.
  - Identify the software criticality level associated with each hazard and control category (**Frequency Level**: Frequent, Probable, Occasional, Remote, Improbable; **Hazard Category**: Catastrophic, Critical, Marginal, Negligible)
  - Match each safety-critical requirement in the software requirements specification (SRS) against the system hazards and hazard categories in order to assign a criticality level to each requirement.
  - Analyze each requirement using the guide phrases. These guide phrases are meant to initiate discussion and suggest possibilities to consider, not to bound the analysis.
  - Document the results of the analysis

Output of Software Architecture Safety and Hazard Analysis

- The products of the requirements hazard analysis consist of the following items:
  - A list of software hazards.
  - A criticality level for each hazard that can be affected by the software.
  - A criticality level for each software requirement.
  - An analysis of the impact on hazards of the software when it operates correctly or incorrectly with respect to meeting each requirement

Input to Software Detailed Design Safety and Hazard Analysis

- The following information should be available to perform the architecture hazard analysis.
  - Preliminary Hazard List
  - Preliminary Hazard Analysis
  - Safety Analysis Report
  - Software Requirements Specification
  - Software Architecture Description
  - Software Detailed Design Description
  - Software Requirements and Architecture Hazard Analyses
  - Trace Matrix, Requirements to Architecture to Detailed Design

Terminology

- **Safety-critical software** is software whose inadvertent response to stimuli, failure to respond when required, response out-of- sequence, or response in unplanned combination with others can result in an accident or the exacerbation of an accident. This includes software whose operation or failure to operate can lead to a hazardous state, software intended to recover from equipment malfunctions or external insults, and software intended to mitigate the severity of, or recover from, an accident.

- A **critical system** is a system whose failure may lead to unacceptable consequences. The results of failure may affect the developers of the system, its direct users, their customers or the general public. The consequences may involve loss of life or property, financial loss, legal liability, regulatory actions or even the loss of good will if that is extremely important. The term safety critical refers to a system whose failure could lead to loss of life, injury, or damage to the environment.