Software Risk Management
A Calculated Gamble

Hans Schaefer
hans.schaefer@ieee.org

How to manage risk
Not only in testing

Hazard and Risk

A Hazard is

Any real or potential condition that can cause injury, illness, or death to personnel; damage to or loss of a system, equipment or property; or damage to the environment.

Simpler .... A threat of harm.

A hazard can lead to one or several consequences.

Risk is

A risk is any variable in your project, that, within its normal distribution of possible values, could take on a value that is detrimental to your project

The expectation of a loss or damage (consequence)

The combined severity and probability of a loss

The long term rate of loss

A potential problem (leading to a loss) that may - or may not occur in the future.

Risk Management

Risk Management is

A set of practices and support tools to identify, analyze, and treat risks explicitly.

Treating a risk means understanding it better, avoiding or reducing it (risk mitigation), or preparing for the risk to materialize.

Risk management tries to reduce the probability of a risk to occur and the impact (loss) caused by risks.

Purpose of Risk Management

- **Anticipate** and **Identify** risk
- **Minimize** the impact / damage / loss
- **Reduce** the probability
- **Monitor** risk areas for early detection
- **Ensure** management awareness of risks

Know your odds!

If you don't actively attack your risks, they will attack you!
(Tom Gilb)

General Causes of Risk

- Lack of Information
- Lack of Control
- Lack of Time

It is impossible, for complex systems, to know everything before it happens.

An example for a failure

Airbus A-340 G-VAEL, Sept 1994

Symptom: Flight management system hung (and lots of other exciting things).

“Please wait...”

Problem still exists Nov 1999. No more info since...

How they handle this in airlines

Pilots are trained for exceptional situations!
Most training is about (potential) trouble.
When trouble occurs, possible solutions are known.
Handling trouble works better then.

They learn from failures!

Risk Management Details

The Risk management Process is a process for proactive prevention: identifying the things that could go wrong, assessing their impact, and determining which potential problems need to be avoided.

Risk management is an activity that must be performed by all levels of the project to ensure adequate coverage of all potential problem areas. Open communication is required to provide all project personnel with the freedom to identify issues without negative consequences to themselves.

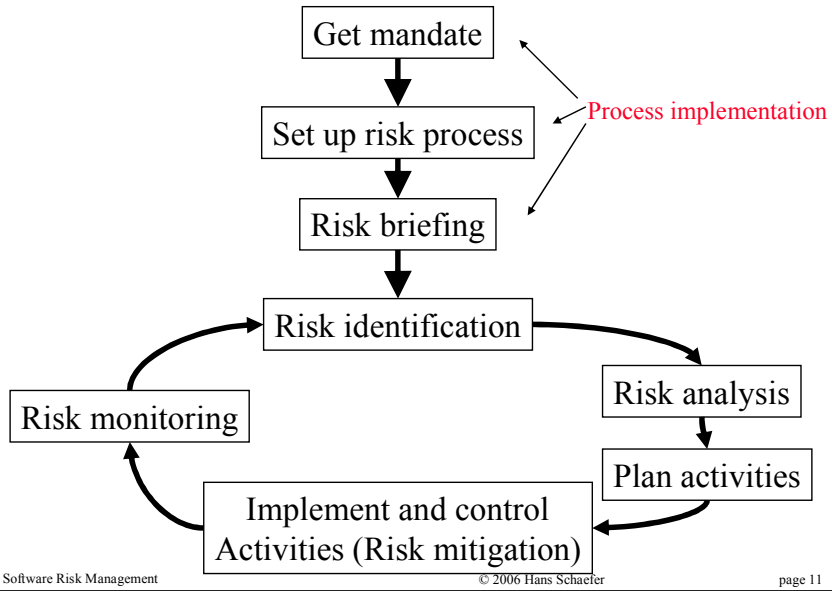
Joint management of risks between acquirer and supplier is necessary to enable identification of the most important risks to the program and to support efficient allocation of mitigation resources. Risk management may make use of the results of other supporting processes such as Problem Resolution, Quality Assurance, and Joint Reviews.

The process consists of the following activities, to be done iteratively:

1. Process implementation
2. Risk identification
3. Risk analysis
4. Risk mitigation
5. Risk monitoring

Step 1: Implementing the Risk Management Process

Risk Management Process



Risk Management Details

1. Process implementation

This activity consists of the following tasks:
Get a mandate. I.e. get resources to implement risk management.

Risk management decriminalizes risk!

Develop a risk management plan. Describe the risk management activities to be done, and the procedures and schedules for these activities. Describe the documentation and reporting requirements, organizations and personnel responsible for performing specific activities, requirements for communicating risks and risk status with other organizations such as acquirers, developers, subcontractors etc. This plan may be part of the project management plan.

Risk briefing: Inform the stakeholders of the project about the risk management plan.

Process Implementation Key success factors

You need a mandate!
- Negative: ”Can do” attitude
- Positive: Project criticality rating

Get all stakeholders involved at start!

Integrate risk management with normal project activities (no separate process!)

Teach, train, support RM!

Keep the process simple!

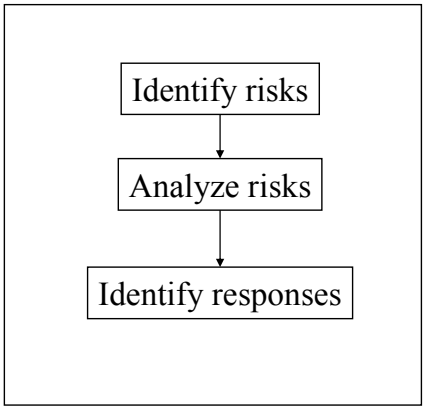
Cyclic process! Risks do not go away, they must be analyzed again and again!

RM Implementation Key Traps

Half-hearted Management Commitment
Everything is floating
No Cyclic Process
People do not accept risk or risk management. No motivation.
Following up ALL risks (more than about 10)
Escalating ALL Risks
    Never cry WOLF when not needed!
Intuitive RM

Step 2: Risk Identification

Risk Identification Workshop agenda



The goal of risk identification

Find possible risks to the project.

Brainstorm!

People have seen risks occur. -> Use your people's experience!

Do not censor!

Result: Unordered stack of "Post it" notes with risks.

Risk Factors

Consider

- **Scope and Quality**
- **Resources**
- **Schedule**

Another way of classifying risks is

- Quality
- Technical
- Human
- Process

General highest level risks

Source: H. Rønneberg, Statoil

User Involvement

Steering group - commitment, decision willingness, involvement

Contract negotiations

Rotten compromises

And more....

Tom DeMarco and Tim Lister, Risk Management during Requirements, IEEE Software, 5/2003

Main interesting points for a test manager:

- Wrong schedules occur often and will put testing under pressure
- If stakeholders cannot agree on requirements, these are often specified vaguely. Test preparation can pull this into the open. Otherwise, it will be found during implementation or testing.
- Personnel loss is common. Some people are lost during the project, thus knowledge must be spread, for example through reviews and documentation.

For a testing project: Where risks come from

- Forward risks (operation of the system -> risk based testing)
- Backward risks (from development: bad quality, delays, uncertainty)
- Risks with the testing project itself (deficient leadership, training, staffing, co-ordination, teamwork)

Quality in software: See ISO/IEC 9126-1 as a checklist

Functionality

Reliability, error handling, safety

Data integrity, security

Usability and user interface

Efficiency, performance, resilience

Maintainability, testability, supportability

Portability, installation, compatibility, interoperability

Technical risks in software:

Component or subsystem interfaces (integration test)

System outside interfaces

Platform maturity

Environment (for example distributed, network)

Data conversion

Capacity

Volume, stress, load

Code coverage

Complexity

Criticality of components

Scope and Quality Risk Remedies

- Unrealistic goals - Prototype, estimate, communicate
- Beyond state of the art - Quantified specification of attributes. Prototypes, Research.
- Complexity of task - Decompose complex tasks into smaller tasks
- New tools and techniques - early use
- Late defect detection - reviews- prototyping
- Assumptions about possible (re)use - check!
- Scope creep - configuration management
- New functionality - Prototype, early warning, later version

Resource Risks

- Loss of products
- Loss of supplier
- Loss of key personnel
- Loss of infrastructure
- Budget overrun
- Budget cut
- Equipment shortage / failure
- Lack of qualification / competence
- Delay in funding
- Technology risks
- High cost tasks
- Other projects or tasks (conflicts)
- Subversive participants

Human risks in software

Familiarity with platform

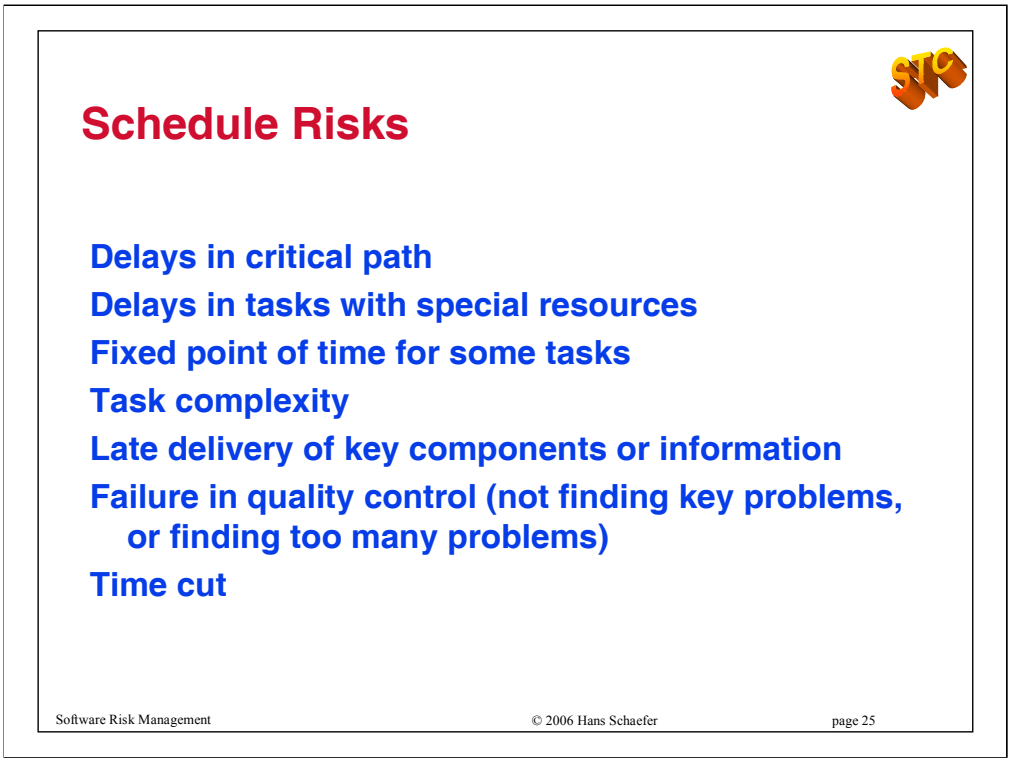
Familiarity with tools, techniques etc.

Geographical spread

Individual temperament

Resource Risk Remedies

- Loss of products - Backup.
- Loss of supplier - Alternative supplier, escrow.
- Loss of key personnel - Buddy system, pair working, people mgt.
- Loss of infrastructure - Double, safeguard, backup, alternative plans.
- Budget overrun - Reserve part of budget.
- Budget cut - List of what is most important.
- Equipment shortage / failure - Have spares, backup, pay for earlier delivery. Check out before project.
- Lack of qualification - Early training, consulting contracts.
- Delay in funding - Check funding points.
- Technology risks - Check how these performed in the past and we will know they will work. Check if technologies are combined in new ways.
- High cost tasks - Which tasks cost most? Which tasks can risk most resources? Which results are the highest investment?
- Conflicting projects - ?
- Subversive participants - Being aware, isolating, confronting such participants, inform higher level management



For software: Process risks

Process familiarity

Process documentation quality

Process maturity

Definition of roles

Time pressure (Schedule risks)

Management style

Schedule Risk Remedies

- Delays in critical path - Time buffer in critical path. Most experienced team members for these tasks. Alternative scheduling. Explore possible earlier delivery of components.
- Delays in tasks with special resources - Which tests can create problems if they run late? Which resources may be a problem? Which tasks are least predictable?
- Fixed point of time for delivery - Duplicate execution of tasks. Independent delivery
- Task complexity - Early feedback
- Late delivery of key components or information - Reduce high sensitivity of plan to the scheduled deliveries.
- Failure in quality control - Make sure test and review WILL find possible key problems. Make sure there are alternatives if something is found to be unacceptable.
- Time cut - Versioning. Early working version.

Some More General Risk Factors

- Inadequate or no development model
- Inadequate or no plan.
- Plan not updated
- Organizational structure inadequate
- Political problems
- Lack of risk management
- No quality policy

Boehm's prioritized top-ten list of software risk items:

Risk item	Risk Management techniques
1. Personnel shortfalls	Staffing with top talent, job matching; teambuilding; cross-training; pre-scheduling; key people; morale building
2. Unrealistic schedules and budgets	Detailed, multisource cost and schedule estimation; design to cost; incremental development; software reuse; requirements scrubbing
3. Developing the wrong software functions	Organization analysis; mission analysis; ops-concept formulation; user surveys; prototyping; early users' manuals
4. Developing the wrong user interface	Task analysis; prototyping; scenarios; user characterization (functionality, style, workload)
5. Gold plating	Requirements scrubbing; prototyping; cost-benefit analysis; design to cost
6. Continuing stream of requirement changes	High change threshold; information hiding; incremental development (defer changes to later increments)
7. Shortfalls in externally furnished components	Benchmarking; inspections; reference checking; compatibility analysis
8. Shortfalls in externally performed tasks	Reference checking; pre-award audits; award-fee contracts; competitive design or prototyping; teambuilding
9. Real-time performance shortfalls	Simulation; benchmarking; modeling; prototyping ;instrumentation; tuning
10. Straining computer-science capabilities	Technical analysis; cost-benefit analysis; prototyping; reference checking

Risk Management Details

2. Risk identification

This activity consists of the following tasks:

Methods should be established for the identification of risks to the program. This should include a statement of risk and a unique identifier, date of identification, and additional context information necessary to fully understand the risk.

There should be some activity like a risk identification workshop, where risks are brainstormed in a non threatening atmosphere.

Similar risks may be combined by abstraction. Too little concrete risks may be detailed.

Risks may have to do with scope, quality, resources, and time.

Step 3: Risk Analysis

Determining Impact, Probability and Surprise

Goal of risk analysis

Find how threatening the risks are.

Weigh the risks.

Make a short list of top risks.

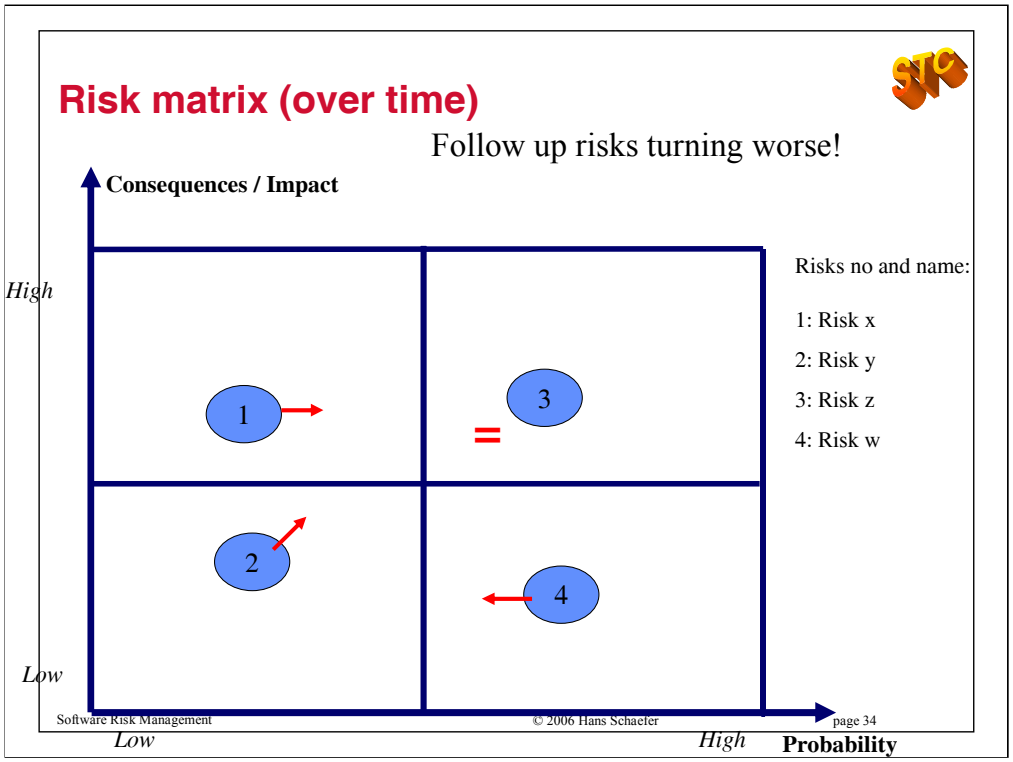
Risk factors

Consequence - How badly does it hurt?

Probability - How likely is it to occur?

Surprise - When will we know it will occur? (not in table down below)

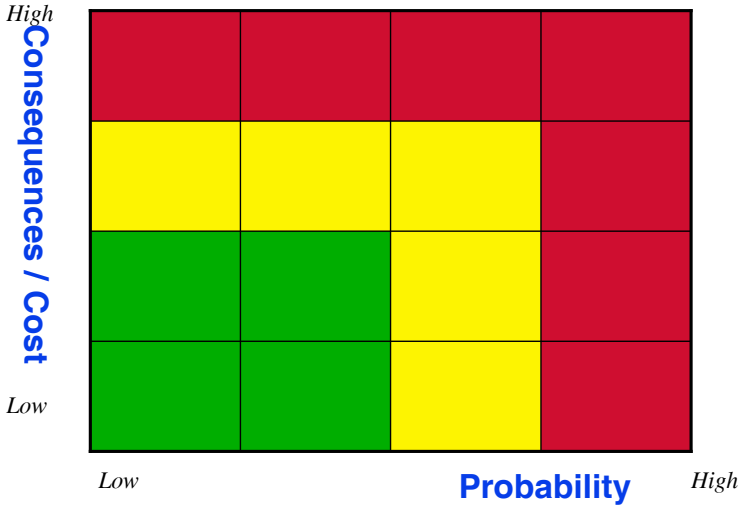
No	Feature / Area	Risk	Consequences H, M or L	Probab. H, M or L	Strategy and actions	When	Responsible



Monitor over time! For example every week.

Risks wandering up and to the right should be monitored more closely.

Risk assessment matrix



Determining Consequences / Impact / Cost

How much will it cost if the risk occurs?
Whom will it hurt (victims)?
How fast will it hurt?
How will it hurt?

From high to low:

- Super-high: Loss of license / company
- High: Impact will likely result in project failure or major renegotiations. (or death, loss > 1MEuro, ...)
- Medium: Difficult to recover fully. Several such risks may doom the project.
- Low: Workarounds are obvious, schedule and cost impacts are minor.

Better: QUANTIFY where possible

Product risks: Things to consider for impact

Critical areas (cost and consequences of failure)

- Catastrophic
- Damaging
- Hindering
- Annoying

Visible areas

- How many users see it?
- How many customers see it?
- How many others, public, see it?

Most used areas

- Unavoidable
- Frequent
- Occasional
- Rare

Can we do without something?

Determining Probability

Do we know anything about the probability?
Or is it all unknown? **UNCERTAINTY! DANGER!**

From frequent to seldom!

- **Frequent (High)**: It would be naive to think it will not happen. ($p > 10^{-1}$ in life)
- **Medium**: It seems to be possible, but not likely. ($10^{-3} < p < 10^{-1}$ in life)
- **Seldom (Low)**: It seems unlikely, but not impossible. ($10^{-3} > p > 10^{-6}$ in life)
- **Improbable**: Everything else

Determining Surprise

When will we detect that the risk will happen?

From high to low!

- High: Risks are detected when they have happened. -> plan reactions to clean up.
- Medium: There is some advance warning, symptoms.
- Low: If this risk is going to happen, it can be seen long before. -> plan preventive actions.

Example: Delays are a low surprise risk. Total failure of some module may be high surprise.

Where the surprise factor is high, monitoring should be tighter, at shorter intervals!

How to classify

Classify into 3 - 5 levels.
Let the pessimist win!

Use a spreadsheet to record and sort.

Why let the pessimist win?

Because everyone on a project tends to be optimist. Testers should “own” a sound pessimism as a weight against this.

Example: Risk classification from ISO 15026

Table 4 - Risk Classification of System Events

Consequences/ Probability	Catastrophic	Hazardous	Major	Marginal	Negligible
Frequent	e	e	e	d	c
Probable	e	e	d	c	b
Occasional	e	d	c	b	b
Remote	d	c	b	b	a
Improbable	c	b	b	a	a
Incredible	b	b	a	a	a

Table 5 - System Integrity Level Assignment

Risk Classification	System Integrity Level
e	E
d	D
c	C
b	B
a	A

Risk Management Details

3. Risk analysis

This activity consists of the following tasks:

Risk attributes of probability, impact and surprise of occurrence should be evaluated. Qualitative and quantitative methods should be used as appropriate to the nature of the risk. It should be evaluated how far risks are avoided by the normal working methods.

Risks should be prioritized to determine their relative importance and to provide a basis for effective use of mitigation resources.

Risks should be classified or grouped into related sets for optimal effectiveness of management and mitigation. Classification should be done using a consistent classification scheme and basis.

Step 4: Risk Mitigation

Goal of risk mitigation

Find appropriate responses to the risks.

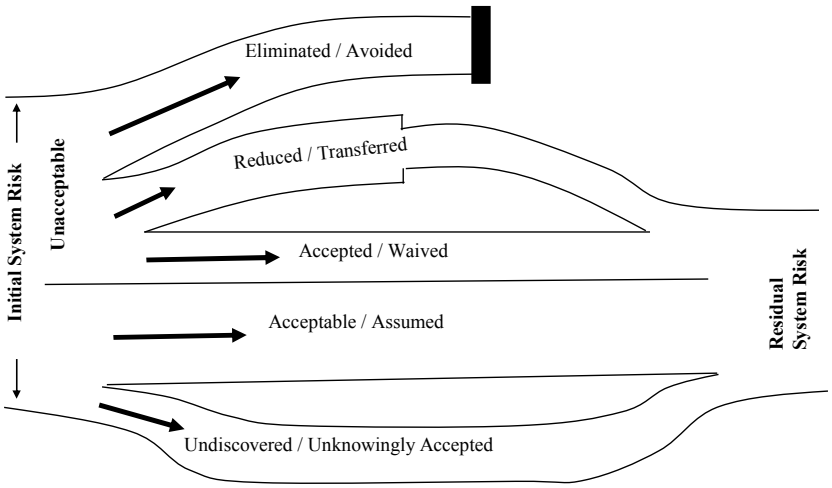
Preventive measures to lower the probability.

Reduction measures to decrease the impact, damage.

Monitoring to lower the surprise factor. Improved early detection.

Discuss every risk!

The risk stream



Possible response to a risk

Accept (do nothing)
Share / transfer (insurance...)

Prevent
Avoid (not prevented risks)
Reduce (not avoided risks)
Plan actions for contingency (not reduced risks)

Testing deals with risks to be prevented.

Discuss measures

Discuss prevention, reduction and monitoring for every risk.

If such measures can reasonably be implemented, the risk may change!

Plan follow up of these measures!

Maybe you need a new risk workshop after this!

Consolidate the sorted risk list.

Top 10 - 20 risks!

Plan follow up

- Assign the top risks an owner.
- Plan regular checking (monitoring), especially for high surprise risks.
- Some risks are ignored.
- Some risks go to external parties (insurance, supplier, customer).
- For some risks, the impact is reduced (prevention of occurrence or of loss).
- For some risks, the alternative actions if the risk occurs, are planned.

Incremental delivery is a great risk mitigation method!

Risk Management Details

4. Risk mitigation

This activity consists of the following tasks:

Risks should be assigned to the appropriate person or organization to be responsible for further management of the risk, including development of mitigation plans, tracking risk and mitigation plan progress, and reporting risk status. Risks that are best managed outside of the organization are transferred (e.g., from a developer to an acquirer).

Each risk is either accepted (closed with no action taken), watched (monitored for significant change), or mitigated. Closed risks are documented with a rationale for acceptance or closure. Watched risks have defined measures, milestones, or metrics defined. Mitigated risks have documented mitigation plans specifying the goal of mitigation, actions to be taken, responsibility for actions, metrics or measures for monitoring progress, contingency triggers and plans (if needed), schedules, and costs.

Monitoring should be done regularly, and each risk to be monitored is assigned a regular interval for monitoring.

Step 5: Risk Monitoring

Risk Monitoring

Integrated into project activities

Top risks checked at regular intervals

Escalation if risk occurs and help needed

-> have a documented escalation process!

Document risk closure

Identify new risks (new cycle)

Who?
How often?

Risk Management Details

5. Risk monitoring

This activity consists of the following tasks:

Risks and mitigation plans are tracked for significant changes in attributes, predefined triggers, thresholds, or events, mitigation plan progress or failure. Status reports are generated as required for reporting to project management, customers, subcontractors, etc., as called for in the risk management plan.

Upon review, risks whose probability or impact has reached a sufficiently low level, or whose mitigation plans have succeeded are closed. Failing or unsuccessful mitigation plans are revised, contingency actions specified in mitigation plans are implemented if contingency triggers have occurred. Joint resolution agreements are required for controlling actions where the risks are jointly visible or jointly controlled.

New risks should be actively searched for at regular intervals or as part of change management.

References

- IEEE Standard 1540-2001: Standard on Software Lifecycle Processes - Risk Management
- ISO 14971 Medical devices - Application of risk management to medical devices
- Boehm, B. Software Risk Management, IEEE Computer Society Press, 1989
- Charette, R. Software Engineering Risk Analysis and Management, McGraw-Hill, 1989.
- Dörnemann, H. Tool-based Risk Management in Requirements Management, CONQUEST 2002, Nürnberg, www.asqf.de, doernemann@computer.org
- Hall, E.M., Managing Risk: Methods for Software Systems Development, Addison-Wesley, 1997.
- Hall, Payson: A Calculated Gamble. In STQE Magazine No 1 +2 / 2003.
- Jones, C., Assessment and Control of Software Risks, Yourdon Press, 1993.
- Rex Black, Managing the Testing Process, John Wiley, 2002. (includes CD with a test priority spreadsheet)
- Leveson, N. G. (1995). Safeware: System Safety and Computers. Reading, Massachusetts: Addison Wesley.
- Randall Rice; Risky Business, A Safe Approach to Risk-based Testing, Better Software Magazine Oct 2006.
- FMEA: Failure Mode and Effects Analysis: **http://www.fmeainfocentre.com/**
- Higuera, R. et al., An Introduction to Team Risk Management, Software Engineering Institute, Carnegie Mellon University, Special Report CMU/SEI-94-SR-1, May 1994. www.sei.cmu.edu.
- J.G. Kontio, Helsinki University: several papers about risk management.
- Rosendahl, E.V., Ton. Performing Initial Risk Assessments in Software Acquisition Projects, in European Conference for Software Quality (ECSQ 2002), Helsinki.
- Stamatis, D.H., Failure Mode and Effect Analysis: FMEA from Theory to Execution, ASQ Quality Press, 2003, ISBN 0-873-895983.
- Heinrich Schettler, “Precision Testing: Risikomodell Funktionstest” Unpublished manuscript, 2005. Heiner.Schettler@t-online.de
- Tom DeMarco and Tim Lister, "Waltzing with Bears:Managing Risk on Software Projects”, 2003.

Links:

- http://www.stickyminds.com
- http://www.pmi.org (Project Management Institute)
- http://www.sra.org (Society of Risk Analysis)
- http://www.risksig.org (Risk Management Special Interest Group)