Prototype Tool Support for SEI Process and Risk Knowledge

Martin S. Feather, John C. Kelly
Jet Propulsion Laboratory,
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109, USA
Martin.S.Feather@Jpl.Nasa.Gov
John.C.Kelly@Jpl.Nasa.Gov

James D. Kiper
Department of Systems Analysis
Miami University
Oxford, OH 45056
kiperjd@muohio.edu

Abstract

We have developed a prototype of tool support for risk assessment that uses selected components of the Software Engineering Institute (SEI) information, specifically: Capability Maturity Model (CMM) process activities, CMM process goals and the SEI taxonomy of software project risks.

This prototype demonstration is hosted on top of DDP, a NASA tool designed for tailoring of mission assurance and evaluating technologies. Using the tool's capabilities, we have cross-linked CMM process activities with the process goals they achieve, and the project risks they mitigate. Populated with this information, we can then use the tool on specific projects to:

- *Navigate rapidly through the SEI classes of information.*
- *Select from the goals and risks those pertinent to the task in hand.*
- *Locate the CMM process activities that will achieve goals and mitigate risks.*

The purpose of this position paper is to convey, at least in part, the status and benefits of this prototype tool.

Contents:

1 SEI information

Since our goal was to produce a prototype for a support tool for risk assessment that can be used on many types of software, it was necessary for us to identify standard lists of software risks and software risk mitigations. The Software Engineering Institute (SEI) is one well-respected source of such information. In particular, the report Software Risk Evaluation Method [Sisti & Sujoe, 1994] presents a taxonomy of software risks. These are organized into three categories: product engineering, development environment, and program constraints. Within each category are risk types and subtypes. For example, the *product-engineering* category has the type *requirements*; requirements risks include stability, completeness, etc.

The SEI's Capability Maturity Model (CMM) for software [Paulk, et al, 1993] categorizes the maturity of the software development process of organizations. Each of the five maturity levels (initial, repeatable, defined, managed, and optimizing) consists of several key process areas (KPA). For example, the KPAs of level 2 are requirements management, software project planning, software project tracking and oversight, software subcontract management, software quality assurance, and software configuration management. Each KPA is, in turn, supported by a few goals and is implemented by a group of activities. We interpreted these activities as risk mitigants. That is, these are activities that will improve the software development process, which has the effect of reducing the risks inherent in software development.

2 Basis for Prototype - NASA's Defect Detection and Prevention (DDP) Tool

NASA's Defect Detection and Prevention (DDP) process [Cornford, 1998] is a method for optimizing the collection of mitigation activities performed on a project. It allows one to perform overall risk management for flight systems. A customized tool has been built to support the DDP process. The work we report on in this paper is hosted on top of this same tool.

The principal elements of DDP are:

- Requirements – the desired goals.
- Risks – things that, should they occur, cause loss of requirements. In DDP, these are referred to as Failure Modes (abbreviated to FMs).
- Mitigations – things that, if applied, will reduce the risks and thereby increase attainment of requirements. In DDP these are classified into categories of Preventions, Analyses, Process Controls and Tests (abbreviated to PACTs).

These elements are cross-linked by:

- Impact – a quantitative measure of how much loss of requirement is caused by an FM.
- Effectiveness – a quantitative measure of how much a PACT reduces the likelihood of a FM.

Since mitigation activities incur costs (e.g., budget and schedule), their selection must tradeoff their costs against their benefits. The main purpose of the DDP tool is to aid users in making this selection judiciously.

3 Population of DDP tool with SEI information

We populated the database of the DDP with the SEI risk and CMM goals and activities as described in section 1. The SEI risk taxonomy clearly fits in the DDP category of *failure modes* (FM). The KPA activities seem to naturally fit as PACT since they can be interpreted as activities that mitigate risk. The goals of each KPA were treated as failure modes or risks. Although this may seem odd at first look, it is apparent that the negation of a CMM KPA goal is a risk. For example, one goal in the requirements management KPA of level 2 is the following:

Goal 1: System requirements allocated to software are controlled to establish a baseline for software engineering and management use.

If negated, this becomes the risk of not allocating system requirements to software so that there is no baseline for software engineering and management use. This negation of these goals allowed a natural link between failure modes (risks) and PACTs (mitigants) as describe in the next section.

Currently, there are very few software development organizations that have attained either level 4 or 5 in the CMM. Level 1 is the initial level that has no KPAs. Thus, it seemed natural to restrict our tools to CMM level 2 and 3. Since the current version is a prototype, we have loaded the CMM level 2 KPA goals and activities. An extension to level 3 would require only the addition of more data to our database.

4 Cross-linking SEI information

As described in section 2, one of the most powerful features of DDP is the links between failure modes and PACTs. Linking information between the SEI taxonomy of risk and the CMM activities does not exist in the literature. This is data that we added to our prototype. The links between CMM activities (PACTS) and CMM goals was fairly obvious. However, the links between SEI taxonomy of risk and the CMM activities was time consuming to generate. Addition of this linking data greatly improves the quality of the prototype. We give the caveat that the links are subjective judgments that require refinement and validation.

5 Tool-supported Navigation and Selection

We believe that there is considerable benefit to be gained from a tool that supports navigation through the SEI (or similar) risk information, and assists in tailoring and selection of that information to the task in hand. For example, there are:

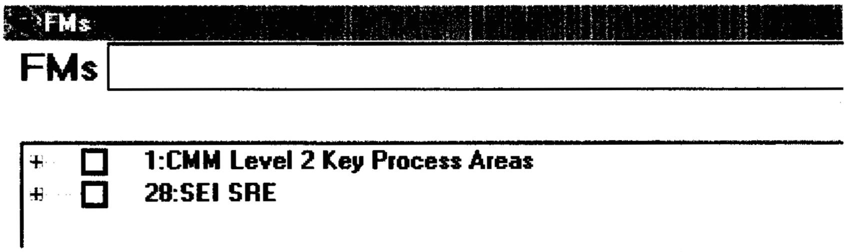
- 20 "leaf" goals in the tree of SEI CMM goals
- 62 "leaf" risks in the SEI risk taxonomy, and
- 62 "leaf" KPAs in the tree of CM level 2 KPAs.

Navigating through this volume of information is inconvenient in the absence of tool support. Selecting an appropriate set of KPAs to match the set of goals and risks for the project in hand is even more problematic without tool support. There are 975 cross-links in the table we populated.

The subsections that follow show (fragments of) screen snapshots taken from our operation of the DDP tool, populated with this information. We intend these screenshots to convey an impression of the ease by which this information may be navigated and applied by using the tool.

5.1 Navigation

The fragment of a screenshot in figure 1 shows the topmost level of "FM" information (i.e., things at risk). The first line, "1:CMM Level 2 Key Process Areas" denotes a tree (currently contracted) of all the level 2 goals. The second line, "28: SEI SRE" denotes another tree (also currently contracted) of the SEI software risks.



FMs

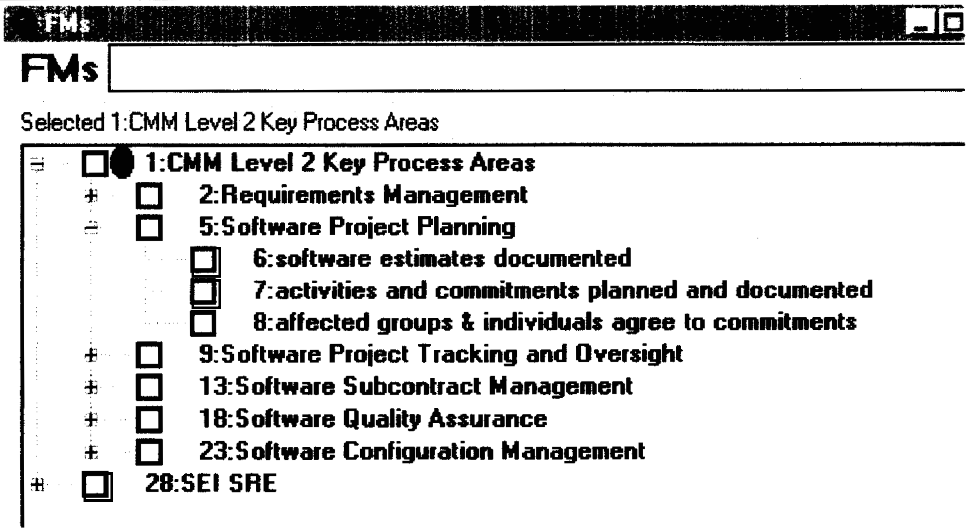
FMs

+ □ 1:CMM Level 2 Key Process Areas

+ □ 28:SEI SRE

Figure 1: Tree of level 2 goals (contracted).

The user can choose to explore either or both of these trees in more depth, by clicking the tiny "+" boxes. This interface is modeled after the file explorer interface familiar to PC users. For example, the fragment of a screenshot of figure 2 shows the tree where the "1:CMM Level 2 ..." node has been expanded, and also the "5:Software Project Planning" node has been expanded. The tiny "+" boxes indicate currently contracted subtrees (e.g., beneath "2:Requirements Management).



FMs

Selected 1:CMM Level 2 Key Process Areas

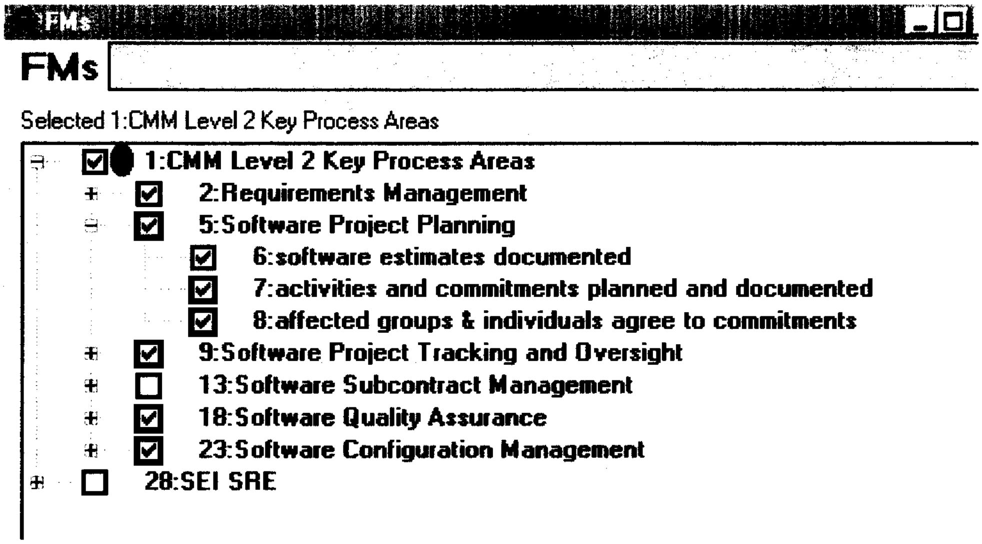
```
- □● 1:CMM Level 2 Key Process Areas
    + □ 2:Requirements Management
    - □ 5:Software Project Planning
          □ 6:software estimates documented
          □ 7:activities and commitments planned and documented
          □ 8:affected groups & individuals agree to commitments
    + □ 9:Software Project Tracking and Oversight
    + □ 13:Software Subcontract Management
    + □ 18:Software Quality Assurance
    + □ 23:Software Configuration Management
+ □ 28:SEI SRE
```

Figure 2: Tree with expanded node.

5.2 Tailoring and Selection

There are two phases of selection – first, selection of the risks that are pertinent, i.e., tailoring the generic risk knowledge to the case-in-hand specifics; second, selection of the mitigations to reduce those risks.

Selecting the pertinent risks is, we think, conceptually easy. A risk should be included unless you can think of a good reason why it is *not* applicable to your project! For example, if the project at hand does not involve any subcontracting, then it is valid to omit the sub-tree of Software Subcontract Management risks. The fragment of screenshot in figure 3 shows such a selection. Most of the boxes to the left of the names now exhibit check marks (the exceptions being the box next to "13:Software Subcontract Management" and next to "28:SEI SRE"). An element is included if and only if its box is checked, and the boxes of all its ancestors in the tree are also checked.



FMs

FMs

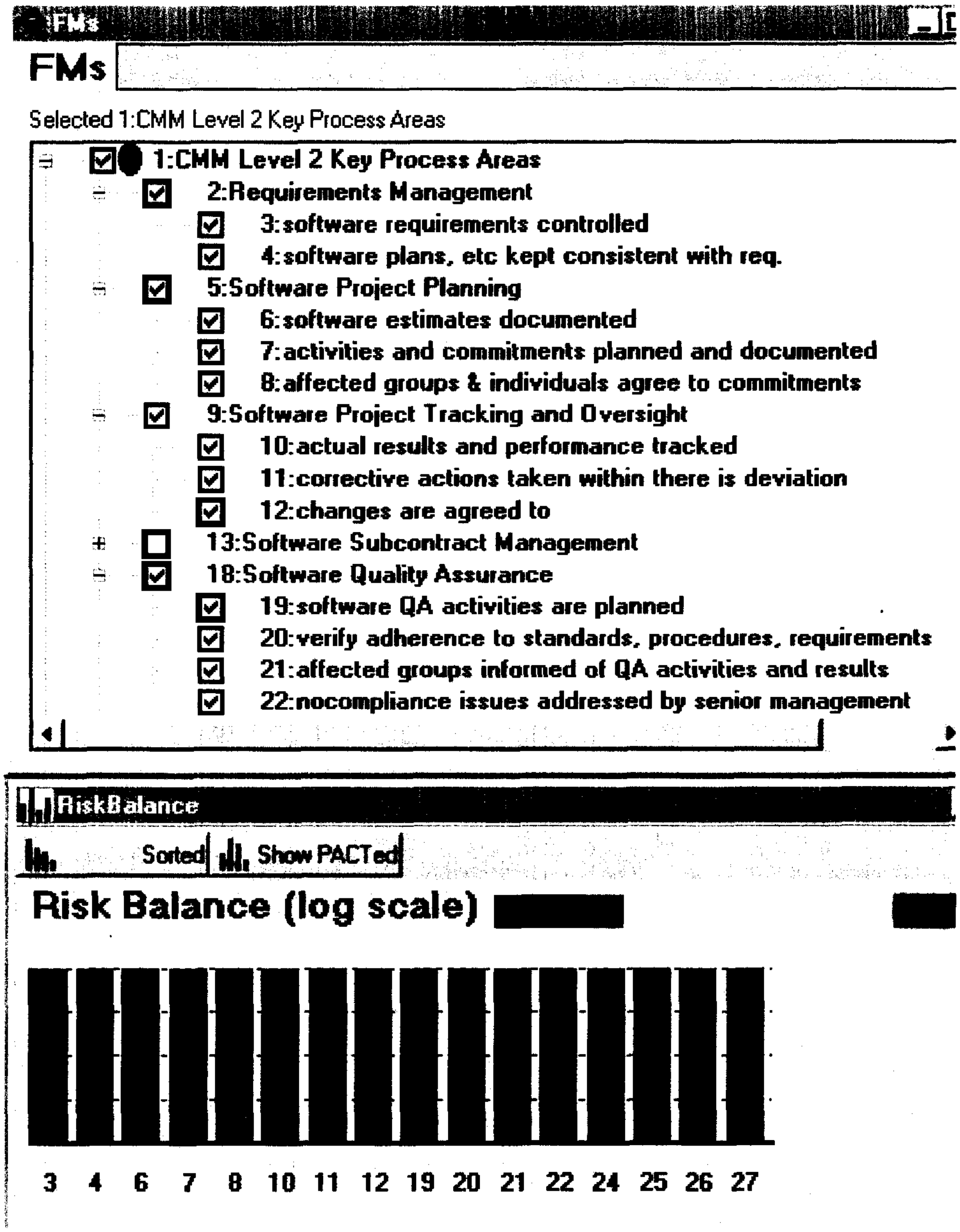
Selected 1:CMM Level 2 Key Process Areas

- ☑ 1:CMM Level 2 Key Process Areas
    - ☑ 2:Requirements Management
    - ☑ 5:Software Project Planning
        - ☑ 6:software estimates documented
        - ☑ 7:activities and commitments planned and documented
        - ☑ 8:affected groups & individuals agree to commitments
    - ☑ 9:Software Project Tracking and Oversight
    - □ 13:Software Subcontract Management
    - ☑ 18:Software Quality Assurance
    - ☑ 23:Software Configuration Management
- □ 28:SEI SRE

Figure 3: Tree with one node deselected.

The tool makes the mechanics of this strategy easy – there is a menu command to select everything in a tree, and thereafter the user can go through and de-select individual items.

Selection automatically drives another display in the DDP tool – a bar chart of risks, where the heights of the bars correspond to the level of risk. For the purposes of this demonstration, we have ranked every risk item as having an equal a-priori impact. The larger screenshot fragment in figure 4 shows a fully expanded (except for Software Subcontract Management) CMM Level 2 Key Process Areas tree, with below it the bar chart of the corresponding risks.



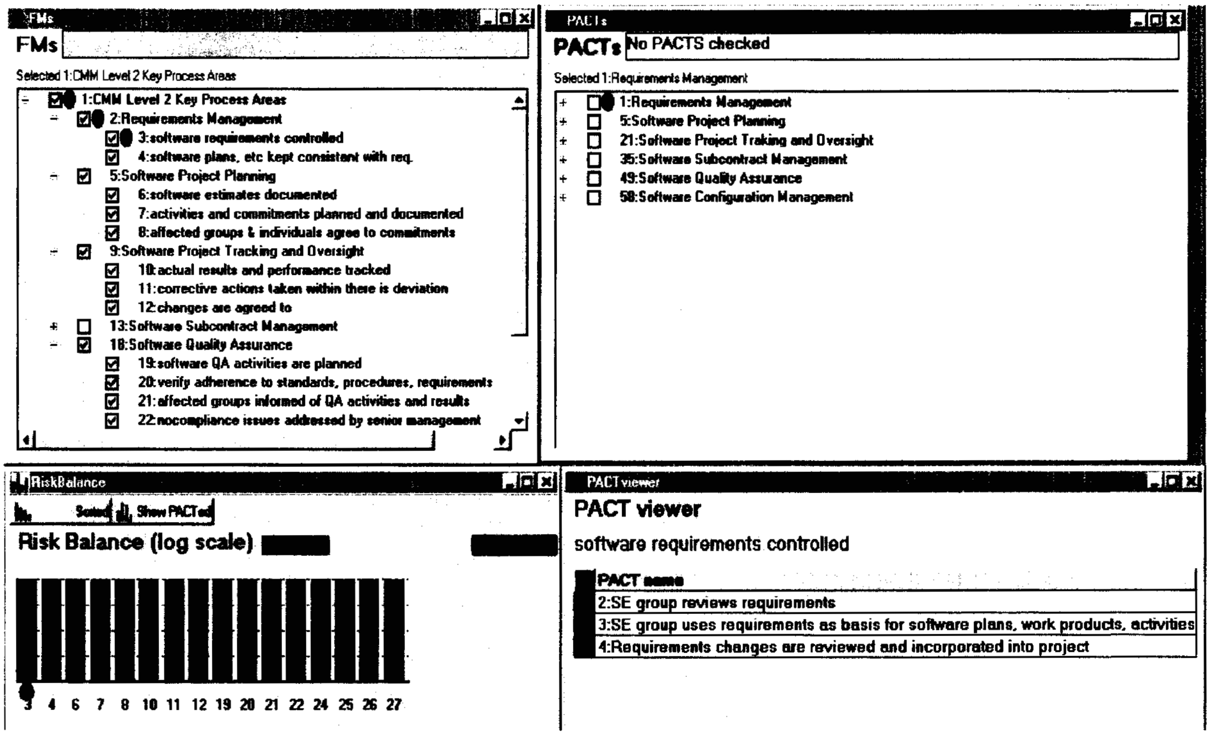
Selected 1:CMM Level 2 Key Process Areas

☑ 1:CMM Level 2 Key Process Areas
- ☑ 2:Requirements Management
  - ☑ 3:software requirements controlled
  - ☑ 4:software plans, etc kept consistent with req.
- ☑ 5:Software Project Planning
  - ☑ 6:software estimates documented
  - ☑ 7:activities and commitments planned and documented
  - ☑ 8:affected groups & individuals agree to commitments
- ☑ 9:Software Project Tracking and Oversight
  - ☑ 10:actual results and performance tracked
  - ☑ 11:corrective actions taken within there is deviation
  - ☑ 12:changes are agreed to
- □ 13:Software Subcontract Management
- ☑ 18:Software Quality Assurance
  - ☑ 19:software QA activities are planned
  - ☑ 20:verify adherence to standards, procedures, requirements
  - ☑ 21:affected groups informed of QA activities and results
  - ☑ 22:nocompliance issues addressed by senior management

Figure 4: Fully expanded tree with bar chart of risks.

Observe that there's a bar for every selected leaf item in the FM tree (the number below each bar indicates the corresponding numbered item in the tree).

The second phase, selection of the mitigations to reduce risks, is the challenging (and fun) part. One way to approach this is to click on a risk bar that you want to reduce: the tool displays a list of all the mitigations that can reduce that risk. For example, the result of clicking on the leftmost risk bar (number 3, i.e., software requirements controlled) is shown in figure 5.



FMs

Selected 1:CMM Level 2 Key Process Areas

- − ☑ ● 1:CMM Level 2 Key Process Areas
  - − ☑ ● 2:Requirements Management
    - ☑ ● 3:software requirements controlled
    - ☑ 4:software plans, etc kept consistant with req.
  - − ☑ 5:Software Project Planning
    - ☑ 6:software estimates documented
    - ☑ 7:activities and commitments planned and documented
    - ☑ 8:affected groups & individuals agree to commitments
  - − ☑ 9:Software Project Tracking and Oversight
    - ☑ 10:actual results and performance tracked
    - ☑ 11:corrective actions taken within there is deviation
    - ☑ 12:changes are agreed to
  - + □ 13:Software Subcontract Management
  - − ☑ 18:Software Quality Assurance
    - ☑ 19:software QA activities are planned
    - ☑ 20:verify adherence to standards, procedures, requirements
    - ☑ 21:affected groups informed of QA activities and results
    - ☑ 22:nocompliance issues addressed by senior management

PACTs No PACTS checked

Selected 1:Requirements Management

- + □ ● 1:Requirements Management
- + □ 5:Software Project Planning
- + □ 21:Software Project Traking and Oversight
- + □ 35:Software Subcontract Management
- + □ 49:Software Quality Assurance
- + □ 58:Software Configuration Management

Risk Balance (log scale)

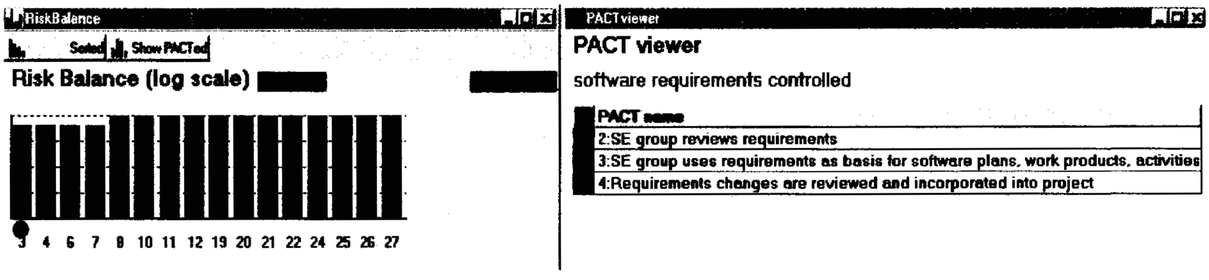
PACT viewer

software requirements controlled

PACT name
2:SE group reviews requirements
3:SE group uses requirements as basis for software plans, work products, activities
4:Requirements changes are reviewed and incorporated into project

Figure 5: Viewing the PACTs that reduce a risk.

The red blob below the bar number 3 indicates it is the focus of attention (corresponding to this, the FMs tree shows red blobs next to the ancestry of that risk element). In the lower right is a window titled "PACT viewer", which is displaying the three SEI process activities that, if selected, would reduce the risk in focus. For example, selecting PACT number 2 "2:SE group reviews requirements" leads to the bar chart shown in figure 6.



Risk Balance (log scale)

PACT viewer

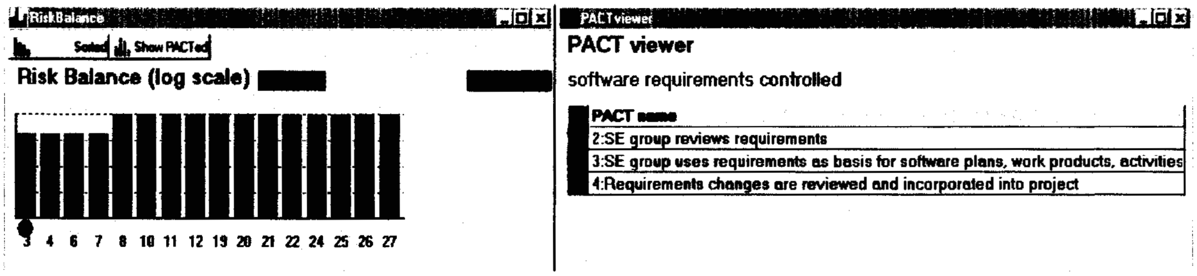
software requirements controlled

PACT name
2:SE group reviews requirements
3:SE group uses requirements as basis for software plans, work products, activities
4:Requirements changes are reviewed and incorporated into project

Figure 6: Risk(s) reduced by application of PACT.

Note that four of the risk bars have decreased in height, not just the one in focus. This is because the selected mitigant reduces the likelihood of all four of those risks.

Selecting a second PACT, e.g., number 4, leads to a further reduction in risk:



Risk Balance (log scale)

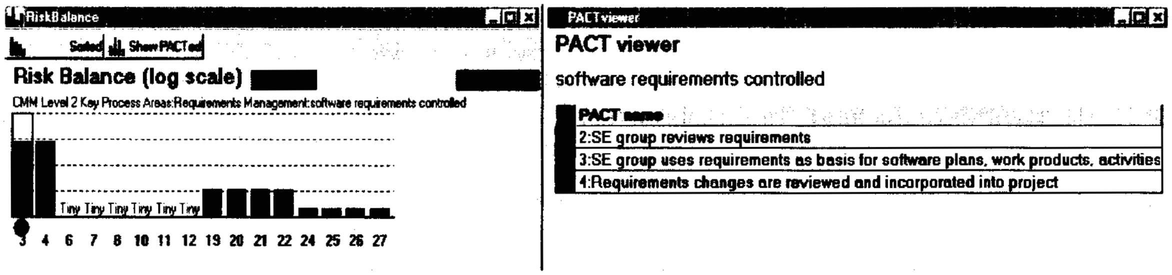
PACT viewer

software requirements controlled

PACT name
2:SE group reviews requirements
3:SE group uses requirements as basis for software plans, work products, activities
4:Requirements changes are reviewed and incorporated into project

Figure 7: Further risk reduction.

If we were to go to the extreme of selecting*all* mitigants, then the resulting risk "profile" is shown in figure 8.



Risk Balance (log scale)

CMM Level 2 Key Process Areas:Requirements Management:software requirements controlled

PACT viewer

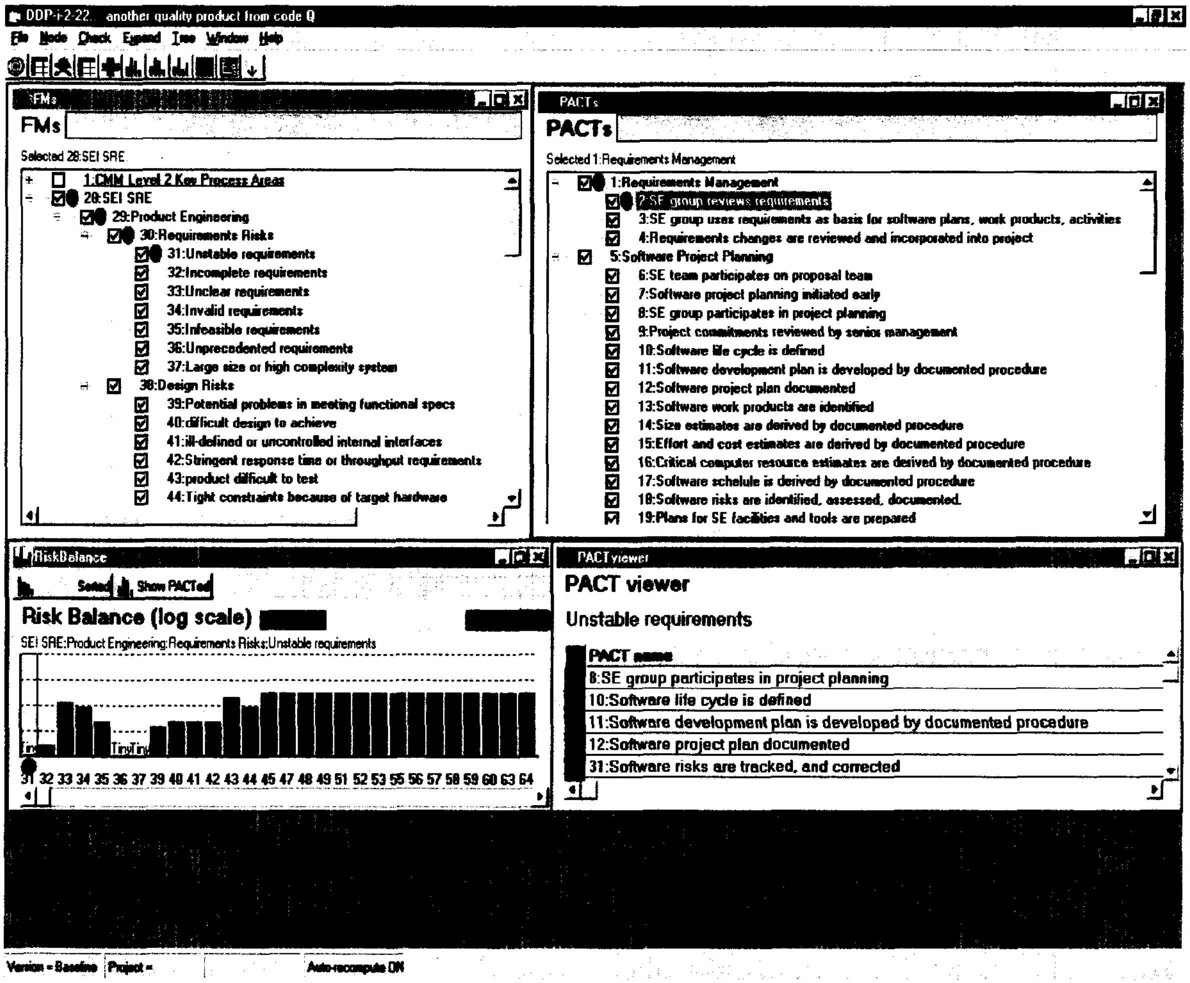
software requirements controlled

PACT name
2:SE group reviews requirements
3:SE group uses requirements as basis for software plans, work products, activities
4:Requirements changes are reviewed and incorporated into project

Figure 8: Application of all PACTs

Most of the risk bars have shrunk significantly, however the leftmost two bars remain as relatively tall. This suggests that there are relatively few SEI activities that reduce these risks, although this observation should be tempered with the realization that our cross-linking of SEI activities against risks is rather tentative (we may well have missed some), and that by default all risks are weighted equally.

A similar process can be followed to select against the SEI risks. The screenshot of figure 9 shows this.



FMs

Selected 28:SEI SRE

- ☐ 1:CMM Level 2 Key Process Areas
- ☑● 28:SEI SRE
  - ☑● 29:Product Engineering
    - ☑● 30:Requirements Risks
      - ☑● 31:Unstable requirements
      - ☑ 32:Incomplete requirements
      - ☑ 33:Unclear requirements
      - ☑ 34:Invalid requirements
      - ☑ 35:Infeasible requirements
      - ☑ 36:Unprecedented requirements
      - ☑ 37:Large size or high complexity system
    - ☑ 38:Design Risks
      - ☑ 39:Potential problems in meeting functional specs
      - ☑ 40:difficult design to achieve
      - ☑ 41:ill-defined or uncontrolled internal interfaces
      - ☑ 42:Stringent response time or throughput requirements
      - ☑ 43:product difficult to test
      - ☑ 44:Tight constraints because of target hardware

PACTs

Selected 1:Requirements Management

- ☑● 1:Requirements Management
  - ☑● 2:SE group reviews requirements
  - ☑ 3:SE group uses requirements as basis for software plans, work products, activities
  - ☑ 4:Requirements changes are reviewed and incorporated into project
- ☑ 5:Software Project Planning
  - ☑ 6:SE team participates on proposal team
  - ☑ 7:Software project planning initiated early
  - ☑ 8:SE group participates in project planning
  - ☑ 9:Project commitments reviewed by senior management
  - ☑ 10:Software life cycle is defined
  - ☑ 11:Software development plan is developed by documented procedure
  - ☑ 12:Software project plan documented
  - ☑ 13:Software work products are identified
  - ☑ 14:Size estimates are derived by documented procedure
  - ☑ 15:Effort and cost estimates are derived by documented procedure
  - ☑ 16:Critical computer resource estimates are derived by documented procedure
  - ☑ 17:Software schelule is derived by documented procedure
  - ☑ 18:Software risks are identified, assessed, documented.
  - ☑ 19:Plans for SE facilities and tools are prepared

SEI SRE:Product Engineering:Requirements Risks:Unstable requirements

PACT viewer

Unstable requirements

PACT name
8:SE group participates in project planning
10:Software life cycle is defined
11:Software development plan is developed by documented procedure
12:Software project plan documented
31:Software risks are tracked, and corrected

Figure 9: PACTs to reduce SEI risks.

6 Long-range goal – ARRT Tool and Techniques

The prototype described in this paper is the first step towards an Advanced Risk Reduction Tool (ARRT).

The Advanced Risk Reduction Tool (ARRT) and associated techniques are being developed to ensure the integrity of the IV&V and QA planning process for NASA projects in a rapidly changing software environment. Lessons learned from software defects and failures on aerospace projects will be effectively integrated into a friendly and professional planning tool that provides a context for making trade-offs from a multitude of choices. A wealth of information is available that makes the traditional manual process of IV&V and QA planning obsolete. A Safety and Mission Assurance Engineer with a software background will be able to negotiate and plan services significantly more effectively using the ARRT tool.

The eventual goal of developing the ARRT support tool is to aid in Independent Software Assessments to ensure that IV&V, QA, and Project customers have:

1. Identified software risks. The risk lists will take into consideration risks associated with software failures on previous NASA and aerospace missions (lessons learned, failure reports, defect profiles, etc.). This includes the identification of software components and intermediate deliverables by level of system criticality.

2. Created an optimized plan that identifies software IV&V, development, and QA activities that mitigate and eliminate software risk for a given NASA project at various times during the lifecycle.

3. Produced consistent cost and schedule risk reduction budget estimates that establish a responsible balance between Faster, Better, Cheaper (FBC) project funding and the safe implementation of software subsystems.

4. Created an equitably negotiated IV&V, Software Development, and QA plan that includes the priorities of the primary stakeholders while maintaining a high integrity program.

5. Produced IV&V, QA, and project plans that are compliant with NASA policy and ISO based Center Software Development Process Descriptions.

A comprehensive support tool will be developed to enable the above five goals to be achieved for current and future NASA projects. The Advanced Risk Reduction Tool (ARRT) and associated technique for this initiative will be designed for software subsystems with an extensible architecture to include techniques demonstrated through research, soon after they prove their effectiveness. A partner in this effort is the NASA Glenn Research Center's "Ask Pete" tool team. Their tool provides estimation and assessment capabilities – see http://tkurtz.grc.nasa.gov/PETE/default.htm for further details.

7 Acknowledgements

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space administration. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.

We especially thank Dr. S. Cornford for allowing us the use of the DDP tool to host our prototype.

This work has been supported by the NASA Software Program (UPN323-08), an agency wide program, managed by the NASA Software IV&V Facility, to promote continual improvement of software engineering within NASA.

8 References

[Cornford, 1998] S. Cornford. Managing Risk as a Resource using the Defect Detection and Prevention process. *International Conference on Probabilistic Safety Assessment and Management*, September 13-18, 1998.

[Paulk, et al, 1993] Mark C. Paulk, Bill Curtiss, Mary Beth Chrissis, Charles V. Weber. Capability Maturity Model for Software, Version 1.1. Technical Report CMU/SEI-93-TR-024, Software Engineering Institute, Carnegie Mellon University, February 1993.

[Sisti & Sujoe, 1994] F. Sisti and J. Sujoe. Software Risk Evaluation Method Version 1.0. Technical Report CMU/SEI-94-TR-019, Software Engineering Institute, Carnegie Mellon University, 1994.