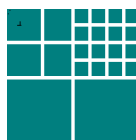
Matching Software Measurements to Business Goals

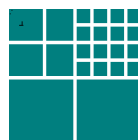
Victor R. Basili

University of Maryland
and
Fraunhofer Center - Maryland



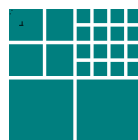
Outline

- Fraunhofer Center Overview
- Importance of Measurement in Software Engineering
- Measurement Issues
- Problems with Establishing a Software Measurement Program
- The Software Measurement Service Approach
- Example Applications
- Status



Center for Experimental Software Engineering, Maryland

- **Fraunhofer Center - Maryland**
  - Software engineering, applied research, technology transfer organization
  - Affiliated with the University of Maryland, College Park
  - Founded in 1998 as part of Fraunhofer USA (1994), a U.S. incorporated, not-for-profit affiliate of Fraunhofer Gesellschaft (1949)
  - Work with government and industrial organizations
- **FC Vision**
  - Apply a scientific/engineering method to software engineering
  - Utilize past results to guide development choices
  - Use organizational learning as the key to improvement
- **Principal Core Competency Areas**
  - Experience Factory, Measurement, Evaluating Technology Maturity, Process Improvement,
  - Reading Techniques, Agile Development, Architectural Evaluation, Risk Management, Process Modeling, COTS-based Development, Security



Importance of Measurement in Software Engineering

Create a corporate memory - baselines/models of current practices
e.g., how much will a new project cost?

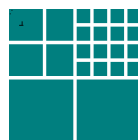
Plan, track and control project development and evolution
e.g., what should happen, is it happening?

Determine strengths and weaknesses of the current process and product
e.g., are certain types of errors commonplace?

Develop a **rationale** for adopting/refining techniques
e.g., what techniques will minimize the problems, change the baselines?

Assess the **impact** of techniques
e.g., does functional testing minimize certain error classes?

Evaluate the **quality** of the process/product
e.g., what is the reliability of the product after delivery?



Software Measurement Issues

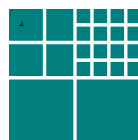
What can we measure?

Resource Data:
- Effort by activity, phase, type of personnel
- Computer time
- Calendar time

Change/Defect Data:
- Changes and defects by various classification schemes

Process Data:
- Process definition and conformance
- Domain understanding

Product Data:
- Product characteristics
  - logical, e.g., application domain, function
  - physical, e.g., size, structure
- Usage and context information, e.g., design method used



Software Measurement Issues

Who are the stakeholders?

There are a variety of stakeholders at multiple levels

e.g., Manager, Customer, User, Organization, Developer

What does each want to know?

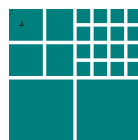
Determines what we measure

But these points of view need to be integrated and linked and interpreted for each viewpoint based on common data

How are the appropriate metrics determined?

There are measurement methods to support metric definition and interpretation

e.g., Goal/Question/Metric Paradigm (GQM), Practical Software Measurement (PSM), Balanced Scorecard (BSC), ...



Software Measurement Issues

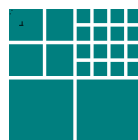
What is needed to support and sustain the activity?

Where is goal definition and data definition and analysis support needed?

- Definition of corporate goals
- Mapping of corporate goals onto software goals
- Mapping of software goals onto measurement goals
- Mapping of goals onto models
- Mapping of models onto existing data
- Interpretation of data based upon goals
- Presentation of data to various stakeholders

How is the measurement process embedded in the organization?

- Organization Structure
  - Integrate many projects into a single measurement framework
  - e.g., Experience Factory, SEPG/Measurement group.
- Data Collection
  - By project aggregate up to central group
  - By central group



Software Measurement Issues

What are the levels of sophistication of goals?
Measurement Capability Maturity

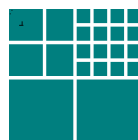
Characterize
- Describe and differentiate software processes and products
- *Build descriptive models and baselines*

Understand
- Explain associations/dependencies between processes and products
- Discover causal relationships
- *Analyze models*

Evaluate
- Assess achievement of quality goals, impact of technology on products
- *Compare models*

Predict
- Estimate expected product quality and process resource consumption
- *Build predictive models*

Motivate/Improve
- Describe what we need to do to control and manage software
- *Build prescriptive models*



Problems Establishing a Software Measurement Program

Defining the right goals

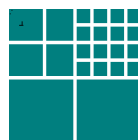
- Tying corporate goals to software goals
- Inheriting software goals from corporate goals
- Identifying the context and temporal aspects of goal definition and achievement

Collecting the right data

- The tension between individual project needs and corporate needs with respect to measures taken
- Maximizing benefits while minimizing costs of data collection and analysis
- Taking maximum advantage of existing data

Defining and Sustaining the measurement process

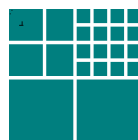
- Creating the right organizational structure
- Getting feedback to projects in a timely fashion
- Maintaining commitment within all organizational levels



Software Measurement Service

- At FC-MD, we have been building a **software measurement service** that
  - uses a decision support system
  - to help an organization define and integrate their top level corporate goals with their software goals
  - map the software goals onto data, maximizing the use of existing data where possible
  - aided by measurement expert

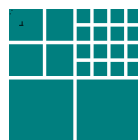
- The contents of this decision support system
  - based upon observations, analysis, and synthesis of actual goals, contexts, and assumptions
  - developed from our interaction with a variety of customers
  - and stored in an experience base



Software Measurement Service

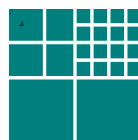
Development strategy

- Analyze, compare, and integrate methods
  - Methods like GQM, BSC, PSM, SPC, Mikko
  - Frameworks like QIP, EF, PROFESS
- Build prototype process for customer engagement
- Document measurement processes and scenarios
- Collect a representative sample of industrial case-studies to
  - Generate sample sets of corporate goals, contexts, scenarios
  - Seed the experience base
  - Test out and evolve the methodology
- Build a prototype set of tools to support the measurement expert



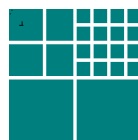
Goal/Question/Metric - GQM

- GQM is a top-down method to define measurements according to stated measurement goals
- Benefits:
  - Makes a visible link from measurement goals to the data collected
  - Creates a detailed measurement plan
  - Gives a model for analyzing collected measurements
  - Involves software developers in measurement definition and analysis
- Weaknesses:
  - Often bypasses top and mid managers
  - Does not create the link to business goals
  - Offers too much flexibility and not enough guidance



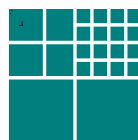
Practical Software and Systems Measurement - PSM

- PSM is issue-based measurement method. It guides project managers to select, collect, define, analyze and report specific software issues. These issues may be risks, problems, new technology, etc.
- Benefits:
  - Project needs initiates the measurement activities
  - Project characteristics guide the metric selection
  - Assesses measurement activities as part of the method
- Weaknesses:
  - Does not create the link to business goals
  - See measurement as project level activity only
  - Can produce too many metrics/ too much data



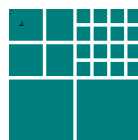
Balanced Scorecard - BSC

- The goal of BSC is to engage everyone to make organization strategy specific and actionable
- Incorporates four measurement views under one management system and tries to balance them: financial, customer, internal business process, and learning and growth measurement.
- Benefits:
  - Strong management focus
  - Links various company aspects under one management system
  - Links measurements to company vision and goals
- Weaknesses:
  - Gives slight or no support for project level measurement definition
  - Operates mainly in top-management level
  - Not software development specific



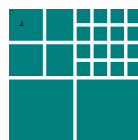
Statistical Process Control - SPC

- SPC aims to control the process through the use of statistical techniques, mainly control charts
- By controlling SPC means keeping the process within its normal performance boundaries thus the upper and lower control limits have to be calculated
- Benefits:
  - Takes business goals as starting point
  - In stable project environment provides a powerful tool to manage process
- Weaknesses:
  - May take place only after firm measurement practices and environment already exists
  - Statistical analysis requires large sample to conduct any reasonable analysis



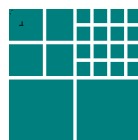
Mikko

- The goal of the MIKKO framework is to be a comprehensive measurement framework. It provides support for data definition, collection and utilization
- MIKKO put forwards a general measurement process and proposes sets methods and tools to be utilized
- Benefits:
  - Provides general guidance for establishing measurement environment
  - Clarifies the status of existing measurement process
  - Stresses planning of utilization of metrics
- Weaknesses:
  - Does not create the link to business goals
  - Operates mainly in a project level



Organization Frameworks

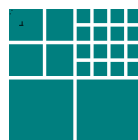
- Quality Improvement Process
  - Aims at applying the scientific method for software
- Experience Factory
  - A learning organization for analyzing, synthesizing and feeding back packaged experiences for projects and corporate needs
- PROFES
  - A derivative of the Experience Factory that supports planning and commitment gathering and creates links between process and product quality



Evaluation and Comparison of Methods

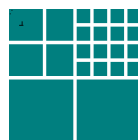
- Qualitative study done by comparing them using the Success Factor Criteria Comparison (SFCC*) method
- SFCC method includes factors found critical for introducing changes in software development environment
  - Developed using SPI case studies and surveys
- SFCC divides success factors to two category:
  - General (Improvement management, Commitment and Cultural Issues)
  - Activity related (according to the PDCA cycle)
- Only if a method emphasises success factor, or incorporates it in the method principles, it is evaluated to fulfil the success factor question and marked ✓

* S. Komi-Sirvio., Development and Evaluation of Software Process Improvement Methods, Doctoral Dissertation in Progress, Espoo. VTT Electronics, 2003. VTT Publications.



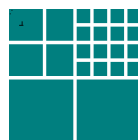
Comparison of Methods

	Measurement Methods					Measurement Frameworks		
Success Factor Questions (Does a method emphasize following)	GQM	SPC	PSM	BSC	MIKKO	QIP	EF	PROFES
Improvement Management								
1. Does method ensure active participation of all affected party?	✓		✓	✓			✓	✓
2. Does method ensure co-operation with software engineers?	✓		✓		✓		✓	✓
3. Does method ensure that training is planned and part of the initiative?			✓		✓			✓
Commitment								
4. Does method ensure commitment of top managers?				✓				✓
5. Does method ensure commitment of middle managers?				✓				✓
6. Does method ensure commitment of engineers?								✓
Cultural Issues								
7. Does method ensure that improved solutions are developed individualistically?	✓	✓	✓	✓	✓	✓	✓	✓



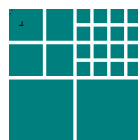
Comparison of Methods

	Measurement Methods					Measurement Frameworks		
Success Factor Questions (Does a method emphasize following)	GQM	SPC	PSM	BSC	MIKKO	QIP	EF	PROFES
Plan								
1. Does method ensure that current status of processes is clarified?	✓	✓			✓	✓	✓	✓
2. Does method ensure that the link between business goals and measurement goals is established?		✓		✓				
3. Does method ensure measurement goals are based on needs and well understood?	✓			✓	✓	✓	✓	✓
4. Does method ensure that detailed measurement plan is generated?	✓		✓	✓	✓			✓
Do								
5. Does method ensure that developed solutions are tested before proposed to a large-scale use in an organization?	✓		✓			✓	✓	✓
6. Does method ensure that practical support is always available for development projects?	✓		✓			✓	✓	✓
Check								
7. Does method ensure that measurement actions are followed regularly using metrics?	✓		✓	✓	✓	✓		✓
Act								
8. Does method ensure sustainability of an measurement initiative?			✓	✓		✓	✓	✓



Measurement Method Process

Software Measurement Service	Method Support
1. Verify Commitment	PROFES
2. Identify Goals	BSC, SPC, PSM, PROFES
3. Characterize Projects	QIP, GQM, MIKKO
4. Develop Goal-Metric Scenarios	
5. Plan Implementation	GQM, MIKKO
6. Implement and Measure	GQM
7. Analyze	GQM
8. Package	EF



Goal Derivation Concepts

Goals

Needs of a particular stakeholder set for prescribing information

Business Goals – goals the organization wishes to accomplish in general,

Software Goals – goals related to the software process or product directly

Measurement Goals – goals that can be made operational

Scenarios

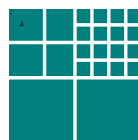
A selected activity path for identifying the right goals and applying the appropriate metrics for the measurement context.

Assumptions

Basic connections made among goal relationships that need to be recorded

Context Factors

Variables in the environment that change the kind of data, the scenarios, ….

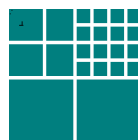


Building Measurement Scenarios

Select the right **goals**:

Identify the relationship between corporate and software **goals**, the **scenarios**, **assumptions**, **context factors**, ….

- Business goal: Reduce product time to market
  - Assumption: Software is on the critical path to product delivery - thus shortening software development time is a reasonable approach
- Related Software Goals: Reduce software functionality, shorten cycle time of individual software activities, trade-off software characteristics for schedule,…
  - Context Factor: Is this a one time/project goal or a long term corporate goal, meant to be sustained? This affects the resources to be allocated, ...
- Selected Software Goal: Shorten Cycle time of individual software activities: perform activities more efficiently, overlap activities, identify new time saving activities, ….
- Selected Software Sub-goal: Perform activities more efficiently: what is the current calendar time of activities, which are using up the most schedule (where are the biggest opportunities for improvement), ...

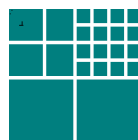


Building Measurement Scenarios: Example 1

Choose the right **scenarios**:

Select the right measurement goals based upon what can you assume about the environment's maturity with respect to measurement

- **Perform activities more efficiently**: what is the current calendar time of activities, which are using up the most schedule (where are the biggest opportunities for improvement), ...
  - **Context Factor A**: Baseline data exists at the activity level
  - **Assumption**: The selected set of projects that form the baseline is relevant to the current situation
- **Scenario A**:
  - 1. Build a schedule baseline by activity,
  - 2. Identify activities that use a major calendar time,
  - 3. Identify opportunities for improvement
  - 4. Apply (Test) out the opportunities for improvement and record the effect on schedule

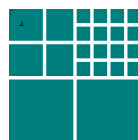


Building Measurement Scenarios: Example 2

Choose the right **scenarios**:

Select the right measurement goals based upon what can you assume about the environment's maturity with respect to measurement

- **Perform activities more efficiently**: what is the current calendar time of activities, which are using up the most schedule (where are the biggest opportunities for improvement), ...
  - **Context Factor B**: No data exists
- Scenario B:
  - 1. Propose explicit hypotheses about baselines, problems, and opportunities for improvement based upon available expertise
    - Assumption: The guesses at the baselines are reasonable and will be updated with real baselines when available
  - 2. Apply (Test) out the opportunities for improvement and record the effect on schedule



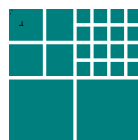
Building Measurement Scenarios

The results from the previous steps provide the information needed for measurement goals (GQM structure)

- Analyze **representative projects** in order to **characterize** them with respect to **calendar time & effort for each phase, activity and in total** from the point of view of the **Corporation**
- Analyze **baseline data** in order to **understand** them with respect to **schedule shrinkage opportunities** from the point of view of the **Corporation**
- Analyze **representative projects** in order to **evaluate** them with respect to **opportunities to eliminate activities or do them concurrently** from the point of view of the **Corporation**
- Analyze **pilot project** in order to **evaluate** it with respect to **the effect of a schedule shrinkage activity** from the point of view of the **Corporation**

Select the right models, metrics, data given the data available

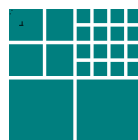
- What data exists? What is the basis for normalizing? Can the data be mapped onto the goals being generated



Industrial Case Studies

Three Test beds for developing and applying methodology

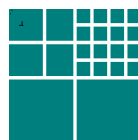
- **NASA Measurement Program**
  - Software ***project measurement*** at the various centers, followed by some high level support for headquarters
  - Show the effect of context variables on measurements taken to define the goals
- **DoD Medical Health Systems**
  - ***Software acquisition*** and SA-CMM
  - Show the relationship between scenarios and goals
- **JPL Mars Science Laboratory**
  - Mars flight and rover systems ***across project and new technology needs*** for system confidence
  - Show that high level goals can collapse on common lower level goals



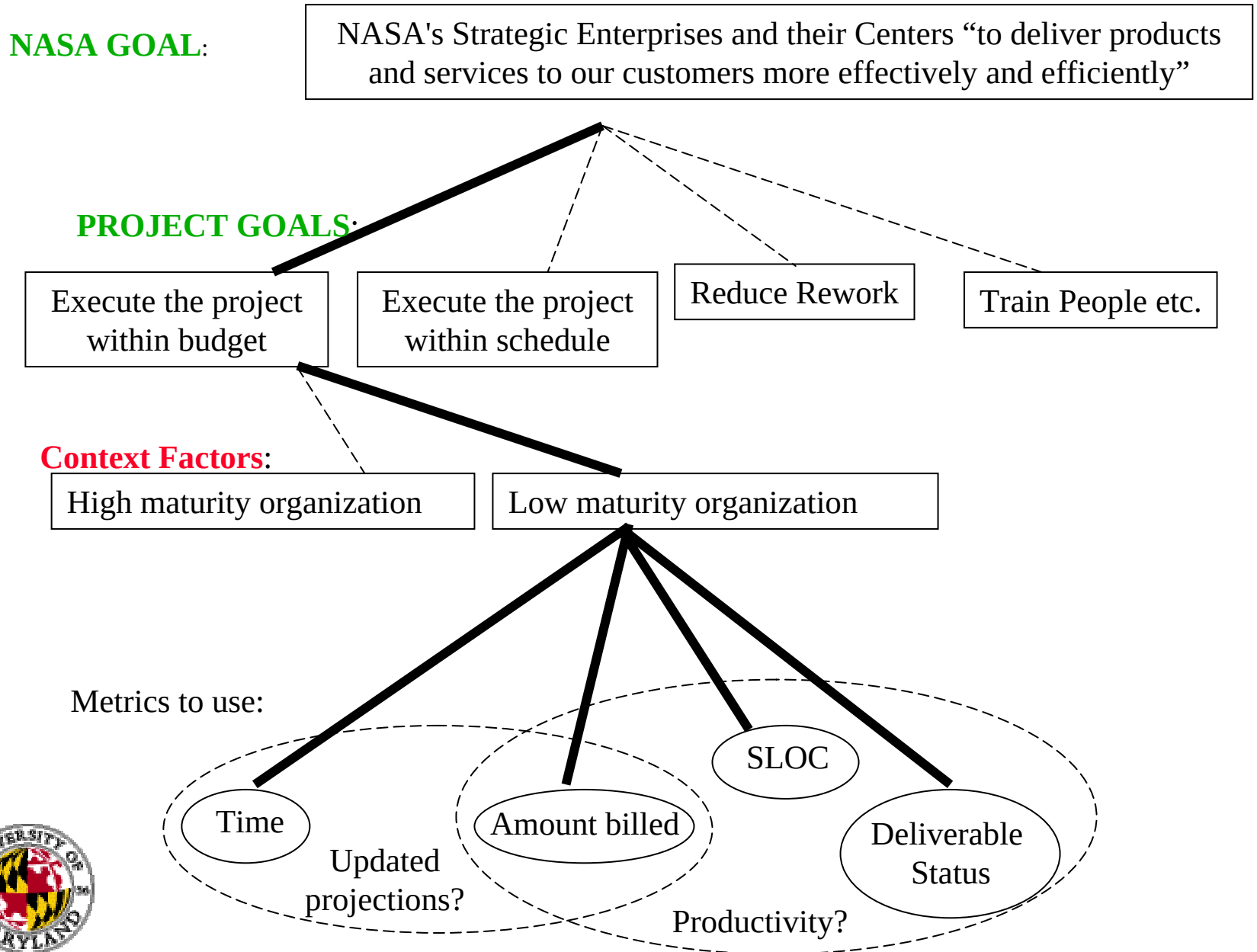
NASA Metrics Selection & Analysis Project

- **NASA Program Goal**
  - "Advance software engineering practices to effectively meet the scientific and technological objectives of NASA"
  - Key objective - establish an agency-wide metrics program (for HQ, Centers, and Projects)

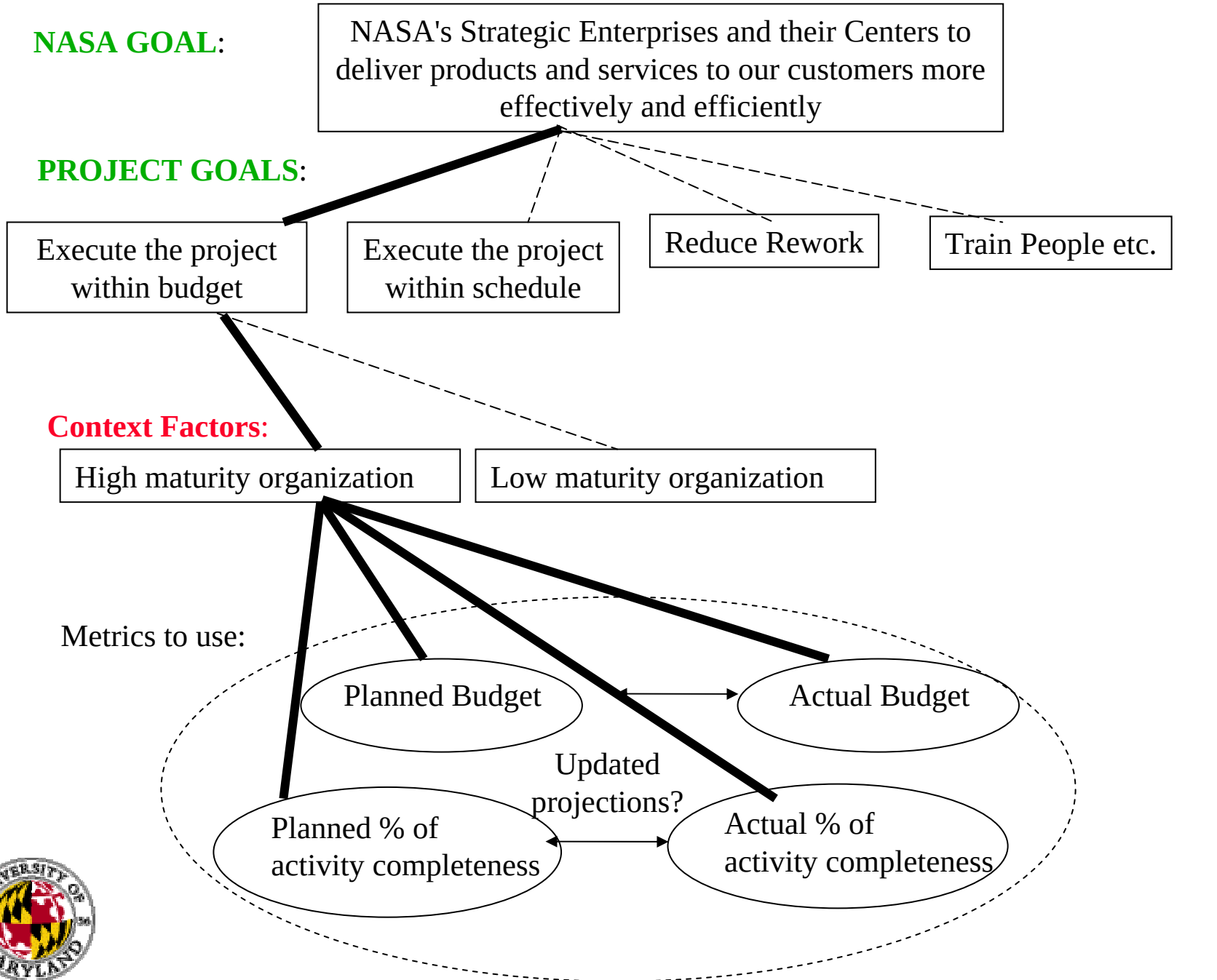
- **Measurement Challenges and Opportunities**
  - Build experience base for hierarchy of project goals, aggregated to headquarters
  - Develop link for project goals to headquarters goals
  - Develop scenarios that work effectively in different project contexts
  - **Recognize the variation in project contexts and provide consistent goals using different models and data, e.g., variation in CMM levels among projects**
  - Make intelligent aggregations
  - Provide guidance and support in deploying the methodology across the projects
  - Develop analysis technique for grouping projects into common "types" based on project characteristics



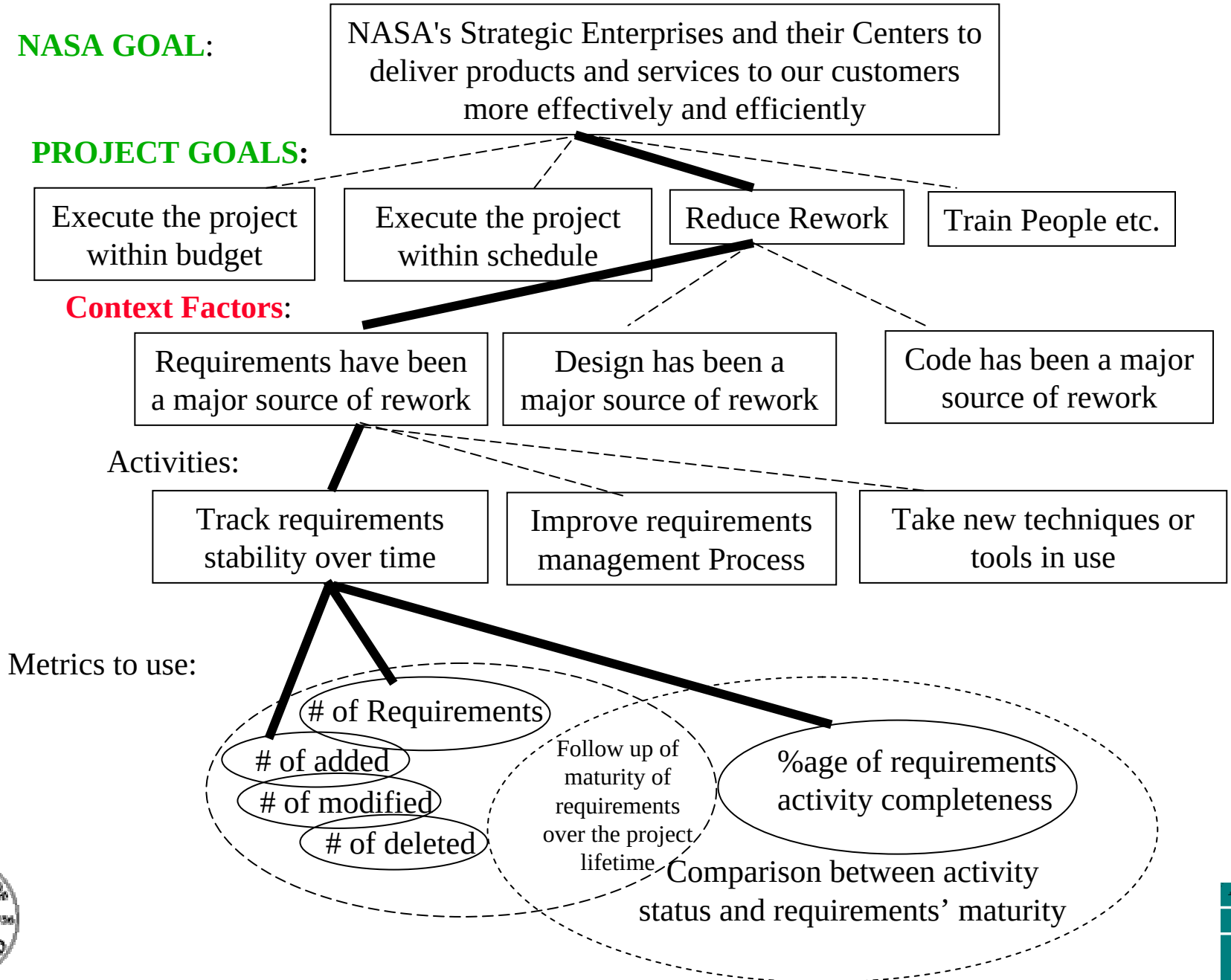
Decision Tree "Execute the project within budget, "low maturity context"



Decision Tree "Execute the project within budget, "High maturity context":



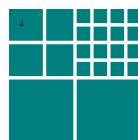
Decision Tree "Reduce Rework":



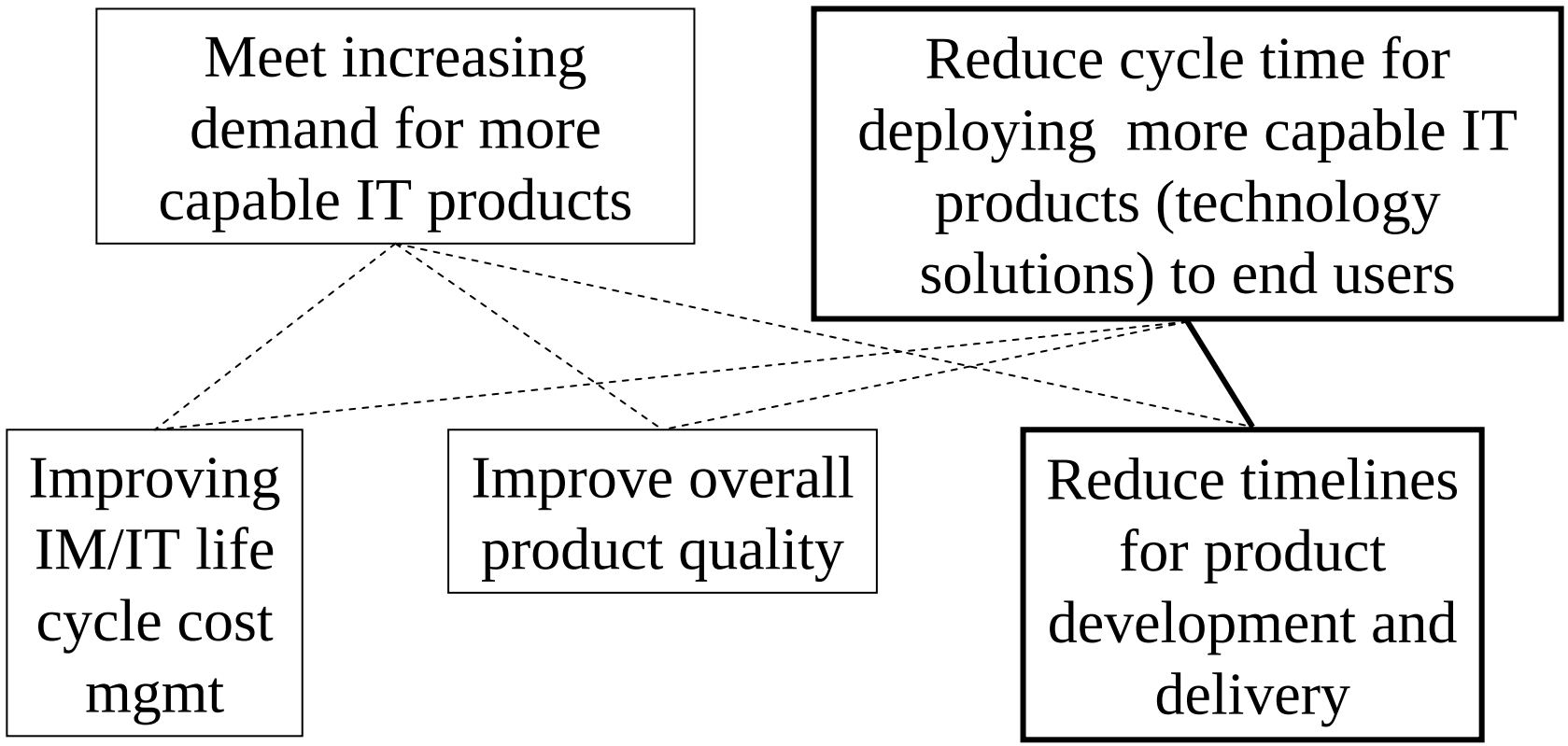
DoD MHS Process Improvement Project

- **Program Goal**
  - Improve and expand their acquisition process using CMM level 2 activities and demonstrate ROI for use of the activities

- **Measurement Challenges and Opportunities**
  - Build experience base for hierarchy of goals for an acquisition organization
  - Define scenarios and templates for acquisition solicitation
  - Identify short term and long term process gains
  - **Develop scenarios that work effectively for SA-CMM**
  - Provide guidance and support for process feedback
  - **Build ROI models for process improvement**



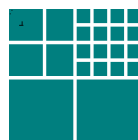
Break-down of Software Acquisition goal(s)



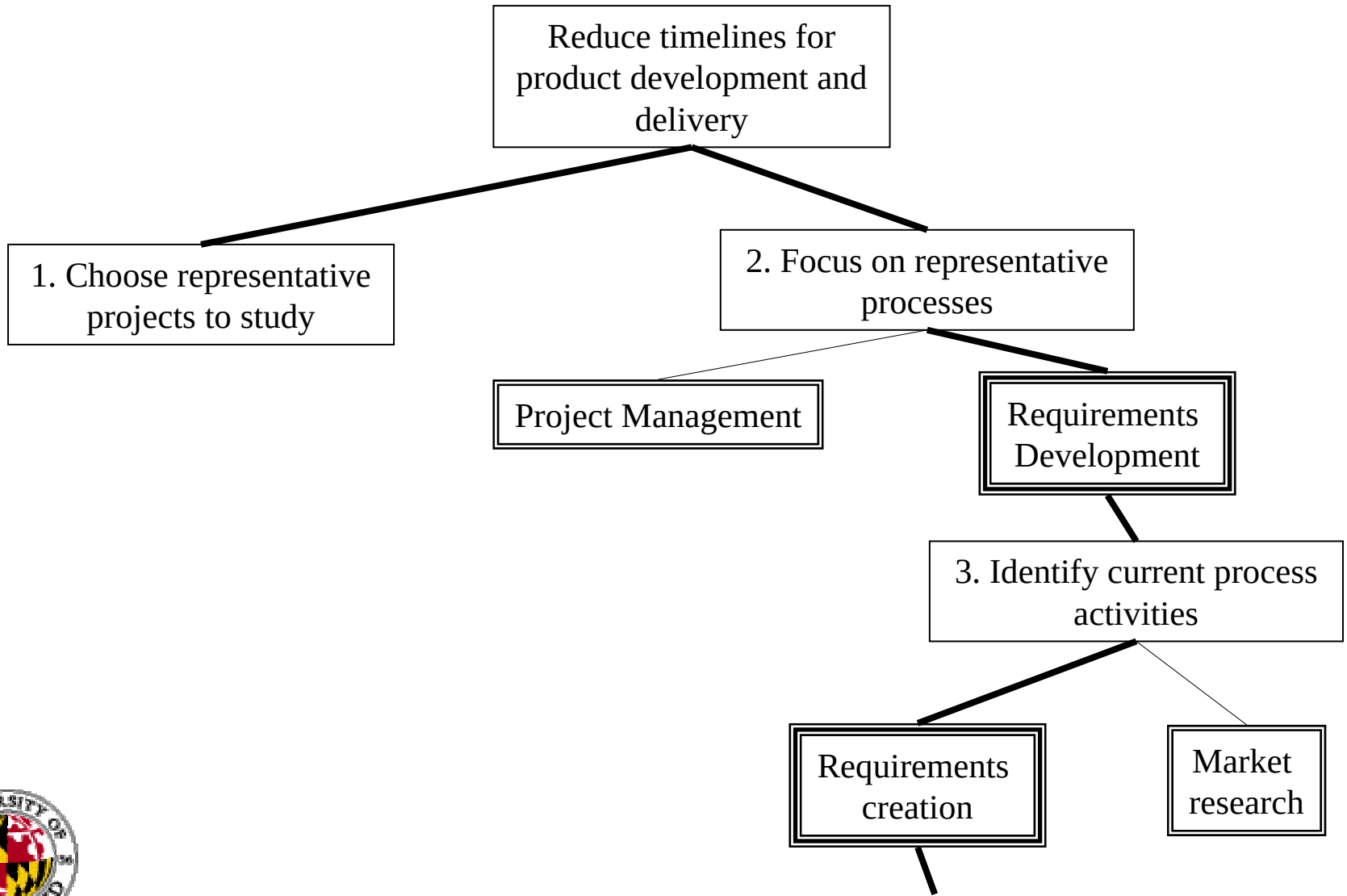
Measuring Reduced Timelines (1)

Scenario for measuring ROI of reduced timelines for product development and delivery:

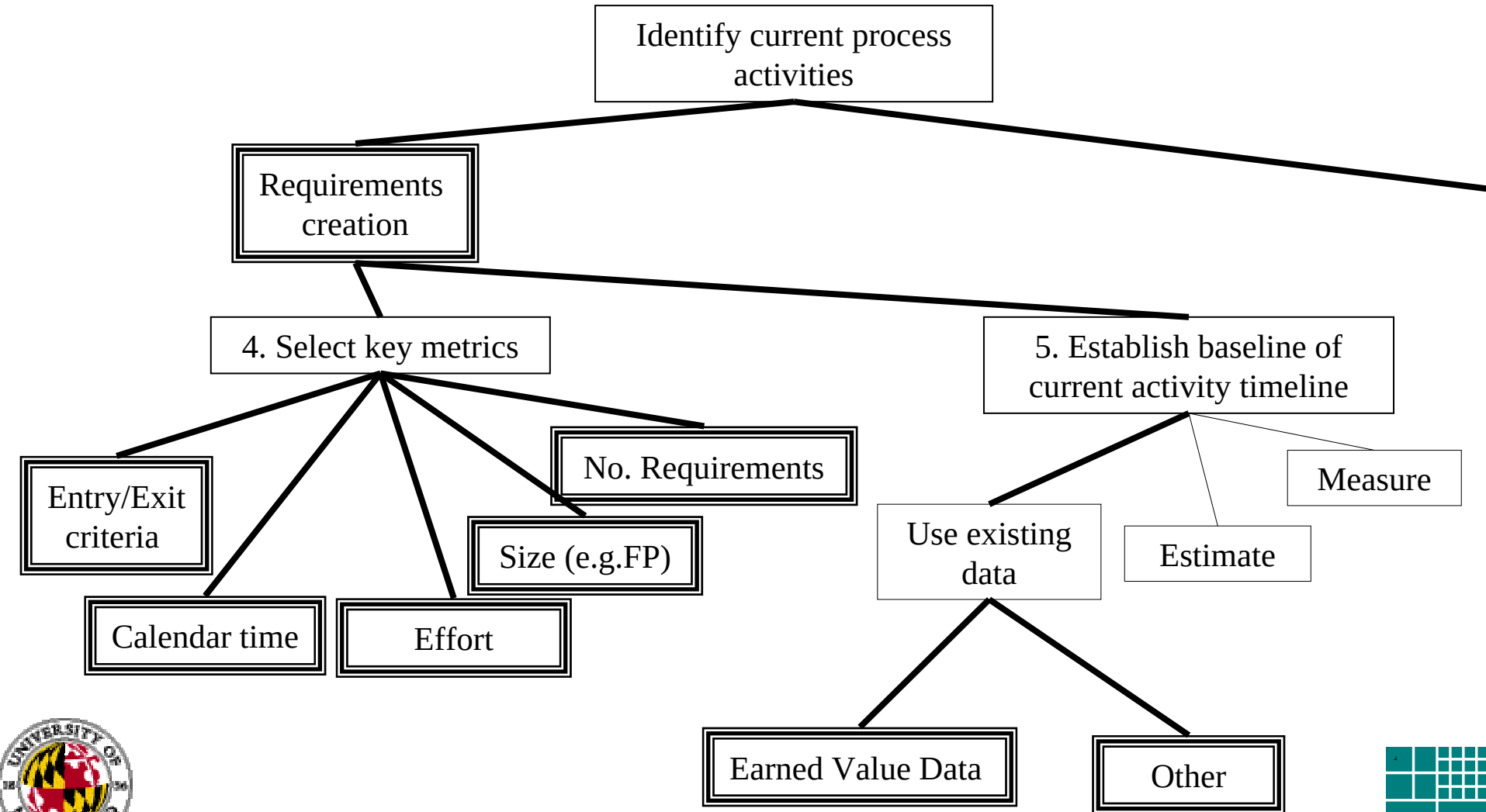
1. Choose representative projects to study
2. Focus on representative processes
3. Identify current process activities
4. Select key metrics
5. Establish baselines of current activity timelines
6. Identify improvement opportunities
7. Estimate ROI of improvements
8. Implement improvements
9. Determine ROI of improvements



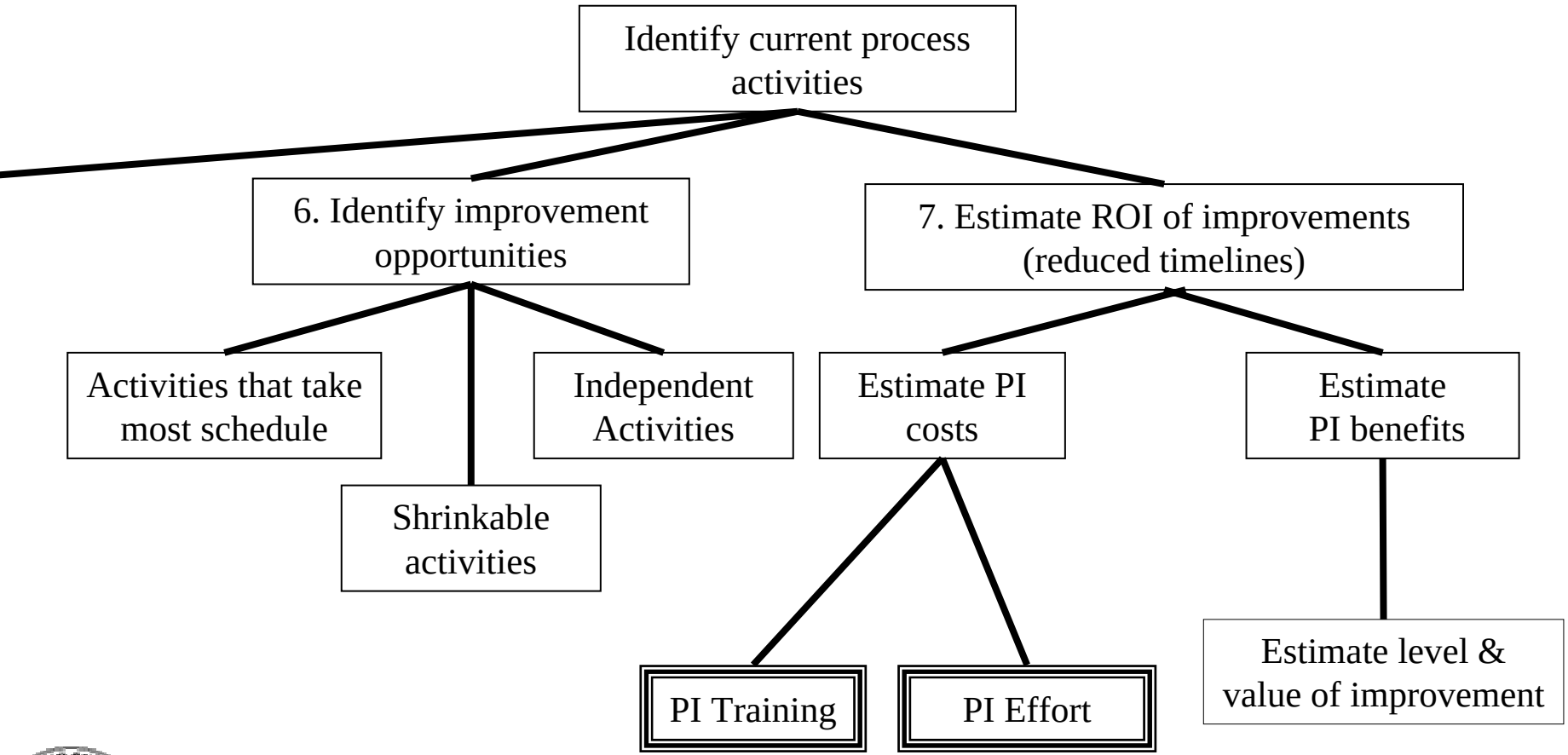
Measuring Reduced Timelines (2)



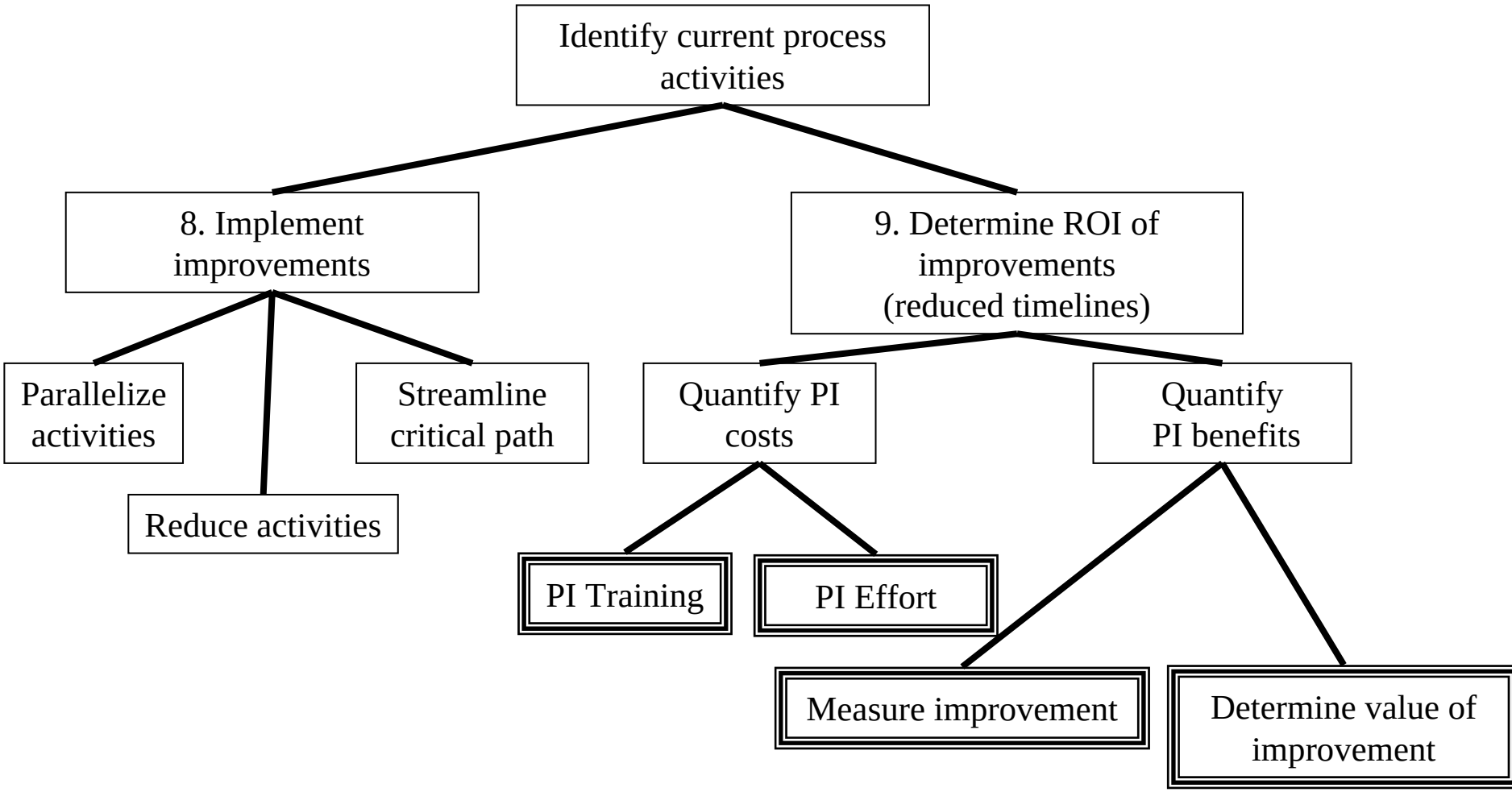
Measuring Reduced Timelines (3)



Measuring Reduced Timelines (4)



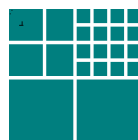
Measuring Reduced Timelines (5)

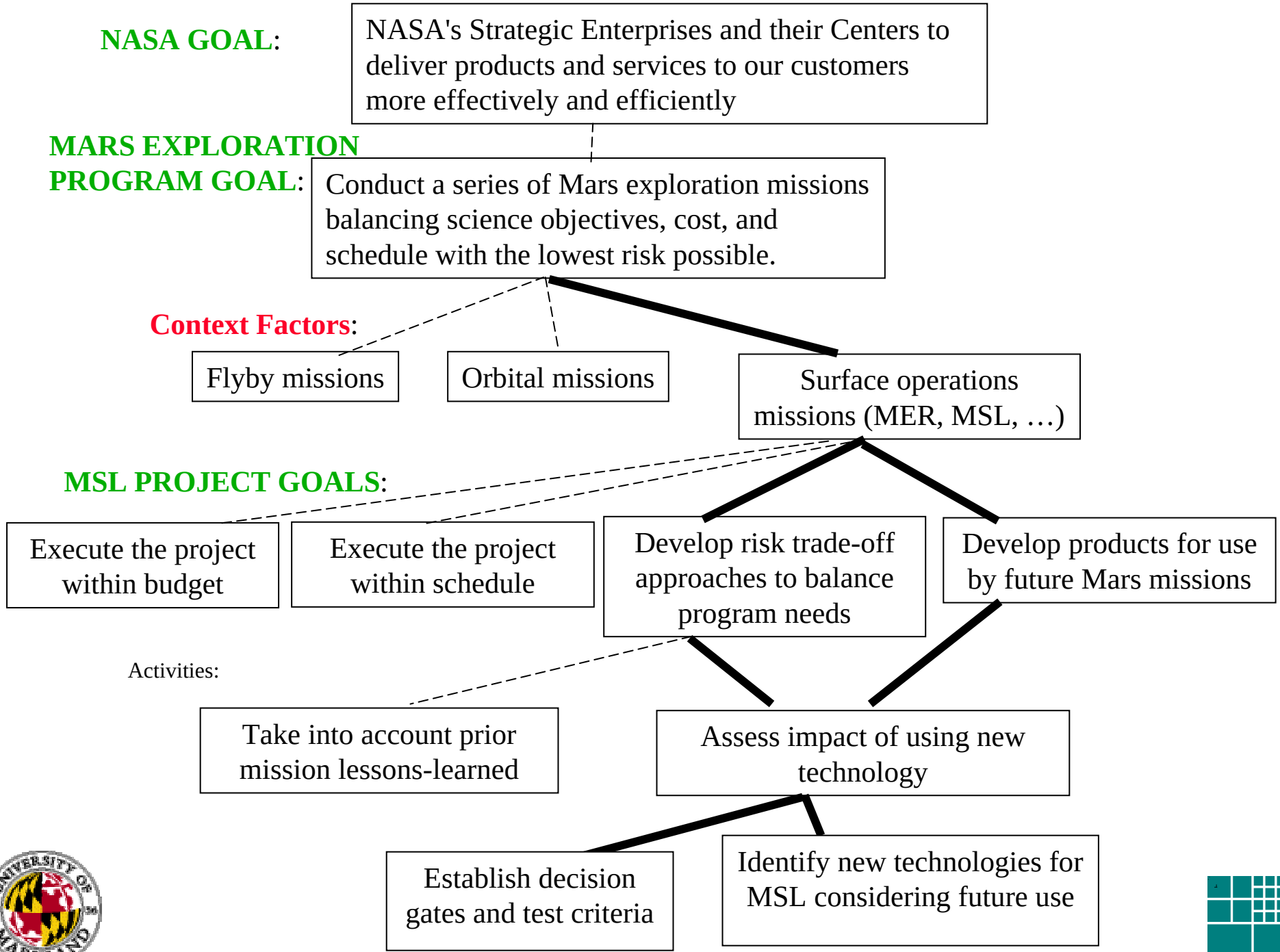


JPL MSL Technology Validation Project

- **Program Goal**
  - Identify V & V activities and measurements that can be used to aid in evaluating the use of new software technology for the Mars Science Laboratory (MSL) at JPL
  - Evaluate the use of autonomous systems to control the mission (flight/rover)

- **Measurement Challenges an Opportunities**
  - Build experience base for hierarchy of goals related to a series of projects
  - Define goals for software technology readiness
  - Define scenarios and templates for confidence in technology use
  - Identify models and measures for confidence building early in the life cycle
  - **Develop goals for assessing V&V strategies**

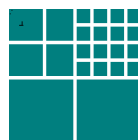



NASA GOAL:
NASA's Strategic Enterprises and their Centers to deliver products and services to our customers more effectively and efficiently
MARS EXPLORATION PROGRAM GOAL:
Conduct a series of Mars exploration missions balancing science objectives, cost, and schedule with the lowest risk possible.
Context Factors:
Flyby missions
Orbital missions
Surface operations missions (MER, MSL, …)
MSL PROJECT GOALS:
Execute the project within budget
Execute the project within schedule
Develop risk trade-off approaches to balance program needs
Develop products for use by future Mars missions
Activities:
Take into account prior mission lessons-learned
Assess impact of using new technology
Establish decision gates and test criteria
Identify new technologies for MSL considering future use

Software Measurement Service

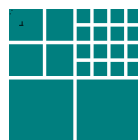
Current Status

- Applying prototype methodology to industrial case studies (using support from individual projects)
- Evaluating and evolving the process
- Integrating BSC and PSM into the method along with GQM
- Analyzing, synthesizing, and generalizing the goals, context, and scenarios from the existing projects
- Building the decision support system tool
- Seeding the experience base with existing information
- Packaging the service
- Identify new customers to expand the domain of goals
- Formalizing the method notation



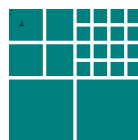
Summary of Key Components for building a software measurement program

- An **experience base** of goals and scenarios that allow for the measurement program to be tailored to specific context variables and assumptions and is based upon experiences with various organizations
- A **method** that takes into account the need for
  - a goal hierarchy that allows goal choices for the needs of a particular organization and stakeholders
  - dependency of goals on one another, i.e., temporal relationships
  - scenarios for identifying clusters, recognizing which types of clusters are needed depending upon environmental constraints
  - mapping goals into existing data sets to maximize information while minimizing data collection
  - the inheritance of data across multiple goals, i.e., mapping the data required from one set of goals onto others
- An **expert** to help set up the measurement program in a the particular organization, including the generation of the goals, measures, data, and analysis



Conclusions

- Building the right measurement program for an organization that deals with its integrated set of global and local needs is a difficult problem
- One size fits all has not been an effective solution
- Software Measurement Service is an attempt at taking advantage of existing methods and addressing these problems



Contributors to the Methodology

- Victor Basili, Fraunhofer Center and University of Maryland
- Kathleen Dangle, Fraunhofer Center
- Patricia Larsen, Fraunhofer Center
- Mikael Lindvall, Fraunhofer Center
- Rose Pajerski, Fraunhofer Center
- Carolyn Seaman, Fraunhofer Center and University of Maryland, Baltimore County
- Seija Komi-Sirvio, Fraunhofer Center and VTT Laboratories
- Marvin Zelkowitz, Fraunhofer Center and University of Maryland

Further information: http://fc-md.umd.edu