Practical Software and Systems Measurement, Users' Group #17

Objective Information for Decision Makers

Workshop #5:
Measuring Software Aspects of Dependability: Revisiting IEEE 982

Workshop Intro & Outbrief
February 25-26, 2016

Rita Creel, Software Engineering Institute

Contents

Workshop Intro

- *Background, objectives, history, & format*

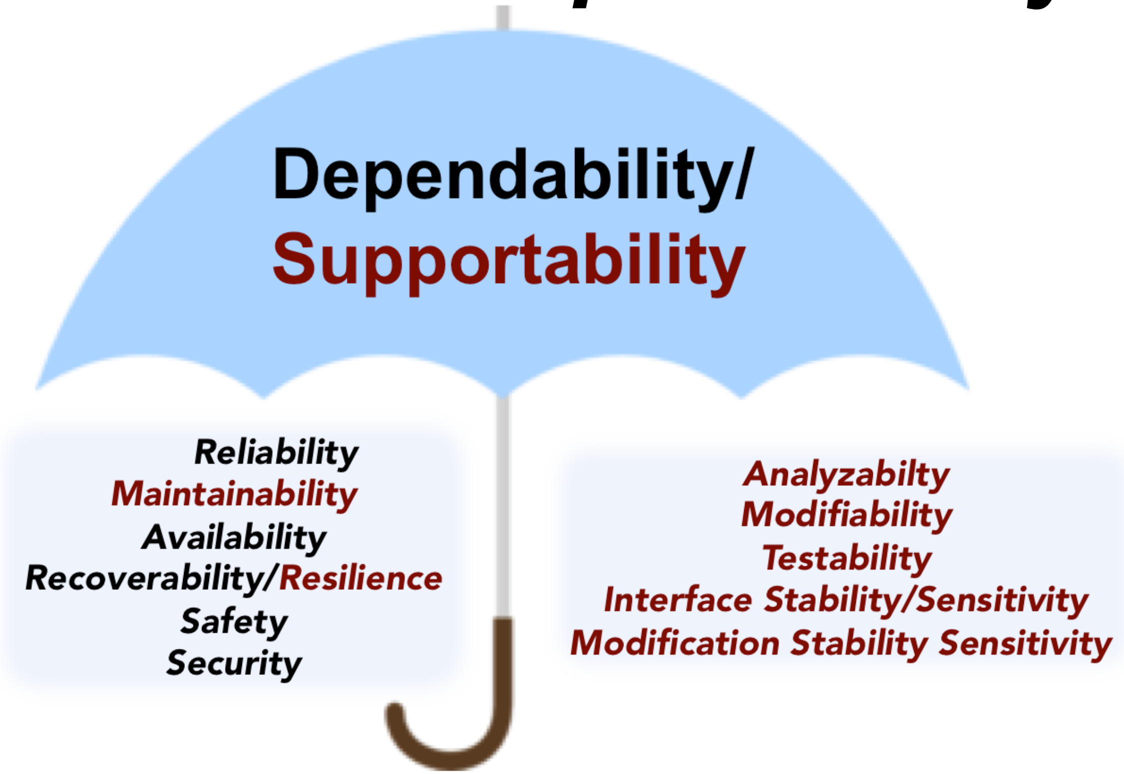
Workshop Outbrief

- *Participants, workshop summary, & next steps*

Additional Materials

- *Workshop aids*

What Is Dependability?



Dependability/
Supportability

Reliability
Maintainability
Availability
Recoverability/Resilience
Safety
Security

Analyzabilty
Modifiability
Testability
Interface Stability/Sensitivity
Modification Stability Sensitivity

Here's one definition – several exist!

Dependability of an item is the ability to perform as and when required.

Dependability includes availability, reliability, recoverability, maintainability, maintenance support performance, and, in some cases, other characteristics such as durability, safety and security.

Source: International Electrotechnical Commission Technical Committee 56 (IEC TC56): DEPENDABILITY, http://tc56.iec.ch/about/definitions.htm

982.1 definition & why it wasn't chosen . . .

Notional: NOT the only or best representation!

*This is **ONE** view of some **EXAMPLE** dependability attributes:* Other attributes, sub-attributes & relationships between attributes & sub-attributes exist.

What Are IEEE 982.1 and P982.1?

IEEE 982.1: Standard for Measures of the Software Aspects of Dependability

- *The purpose of the standard is to provide a set of measures for software quality attributes that result in dependable execution of the software*
- *Initially released in 1988 and last revised in 2005*
- *Decision made in 2014 to revise (vs. reaffirm or withdraw)*

IEEE P982.1: Project to revise IEEE 982.1 to

- *increase the standard's applicability, visibility, and use in light of changes in software/system engineering & management.*
- *advance software measurement practice for quality attributes essential to dependability.*

The P982.1 Working Group welcomes your ideas and support!

Objectives of the Workshop

General objectives are to:

- *Solicit PSM contributions to* ***advance dependability measurement practice*** *for software-reliant systems.*
- *Offer participants an opportunity to play a role in efforts to* ***mature software engineering as a profession****.*

Today's Objectives are to:

- *Elicit* ***questions, recommendations, challenges, successes & ideas*** *related to measuring software aspects of dependability.*
- *Discuss barriers & enablers regarding* ***resources, effort, and expertise*** *needed to establish, measure, & sustain software dependability attributes—and technical, programmatic, & mission risks of not doing so.*
- *Document workshop results in an* ***outbrief*** *and share them with PSM UG attendees and the IEEE P982.1 WG.*

Workshop Format–Agenda

(Topics and order are flexible—and we'll likely iterate.)

Introductions and a question*: Why is this topic of interest to you?*

Discussion the definition of dependability as it relates to software's role.

Briefly review and discuss what exists today.

- *Topics and content of the current (2005) version of 982.1, and related standards and guidebooks (e.g., ISO 250xx, OMG/CISQ)*
- *Scope of the IEEE P982.1 revision effort: How have expectations about software changed since the last revision? What gaps need to be filled?*

Collect participant questions, recommendations, challenges, successes & ideas: What's needed and how can we supply it?

Respond to the following questions:

- *What does it take to measure software aspects of dependability in different contexts?*
- *How can we affordably measure software aspects of dependability throughout the system life cycle?*

Determine next steps and prepare workshop outbrief and evaluation.

Workshop Background

PSM history in this area

- *The ICM* table includes many measures applicable to dependability across the seven PSM information categories.*

  *ICM: Information Category – Measurable Concept – Potential Measure

- *PSM working groups have dealt with many topics related to dependability (security, quality, technical debt, architecture, . . .)*

Where we're heading

- *We're revising IEEE 982.1, the standard for software aspects of dependability, attending to both organizational and technical implementation concerns.*
- *We'd like to use/refine/develop PSM measures to support the standard.*

Issues, questions, and topics

- *How can we select, define, & explain the set of software attributes that characterize dependability?*
  - *How can we measure these attributes throughout the lifecycle?*
  - *Which should be implemented broadly? Which cab we affordably implement today?*
- *How can IEEE P982.1 and PSM advance dependability measurement practice?*

Intended Output

- *Workshop outbrief with discussion results (questions, recommendations, challenges, successes, and ideas)*
- *Dependability measure inputs to IEEE P982.1 revision effort, due to complete December 2018*
- *PSM participant engagement as the P982.1 effort moves forward*

Post-Workshop Activity

- *Status/progress of P982.1 WG will be posted on the Public wiki, https://ieee-sa.imeetcentral.com/p9821wgpublic/*
- *Updates will be provided at subsequent PSM events – with a follow-on workshop, if there's interest*
- *If you'd like to join the WG or stay informed about P982.1's progress, please send me email at rc@sei.cmu.edu*

OPTIONAL Prep

Visit https://ieee-sa.imeetcentral.com/p9821wgpublic/

Bring the following:

- *Your **experiences, frustrations, hopes, disappointments, successes, and ideas** with respect to specifying and measuring software's contribution to system dependability.*
- *Relevant **research or practical frameworks, methods, and tools** for specifying software-related dependability requirements and measures.*
- *Your **thoughts on key barriers** when it comes to practical, affordable dependability measurement **and how to overcome them**.*

Workshop #5:
Measuring Software Aspects of Dependability: Revisiting IEEE 982.1

Workshop Outbrief

Outbrief Topics

Participants & Review of Objectives

Dependability as a construct

History in this area

Discussion topics & summary

Conclusions & recommendations

Next steps & how you can participate!

Workshop Participants

- *Sean Brady*
- *Rita Creel*
- *Joe Dean*
- *Marc Jones*
- *Tori Shu*
- *Simon Lemmo*
- *Bob McCann*
- *Arlene Minkiewicz*
- *Greg Niemann*
- *Pete Pizzutillo*
- *Scott Schorn*
- *Tori Shu*

Objectives of the Workshop

General objectives are to:

- *Solicit PSM contributions to **advance dependability measurement practice** for software-reliant systems.*
- *Offer participants an opportunity to play a role in efforts to **mature software engineering as a profession**.*

Today's Objectives are to:

- *Elicit **questions, recommendations, challenges, successes & ideas** related to measuring software aspects of dependability.*
- *Discuss barriers & enablers regarding **resources, effort, and expertise** needed to establish, measure, & sustain software dependability attributes—and technical, programmatic, & mission risks of not doing so.*
- *Document workshop results in an **outbrief** and share them with PSM UG attendees and the IEEE P982.1 WG.*

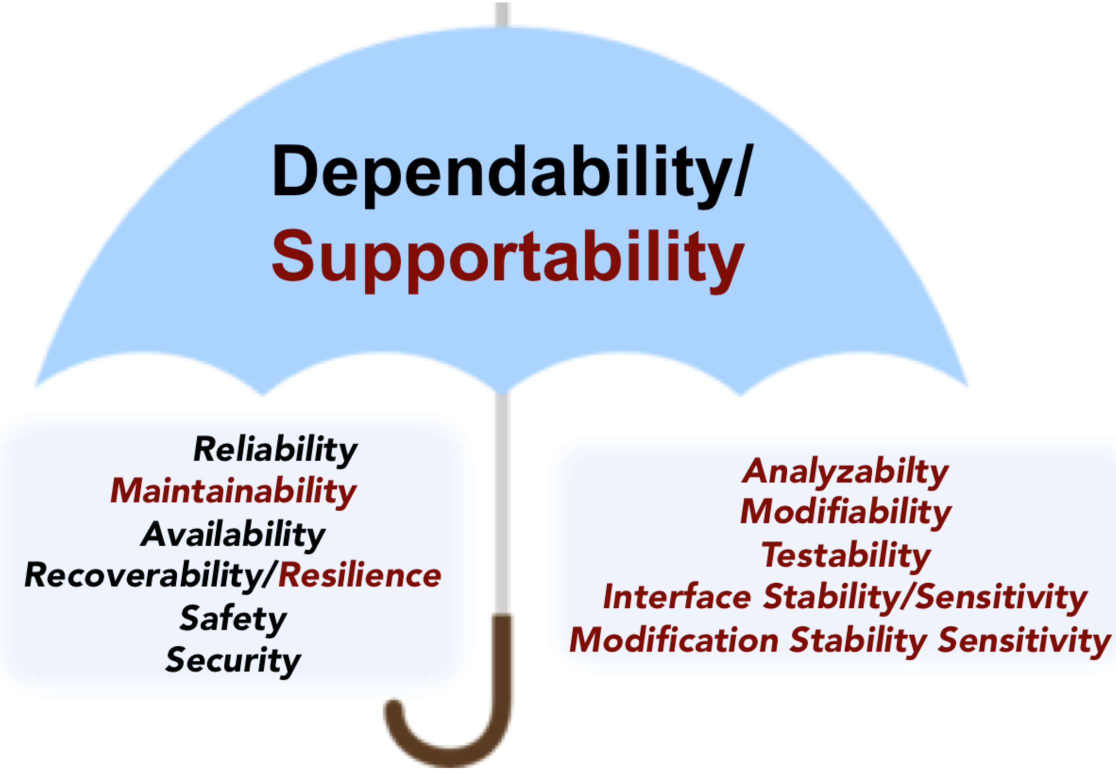
Defining Dependability

… especially as it relates to software.

Many

- *Frameworks*
- *Attributes*
- *Definitions*
- *Ideas about hierarchical and lateral/peer relationships*

What Is Dependability?



*This is **ONE** view of some **EXAMPLE** dependability attributes:* Other attributes, sub-attributes & relationships between attributes & sub-attributes exist.

Here's one definition – several exist!

Dependability of an item is the ability to perform as and when required.

Dependability includes availability, reliability, recoverability, maintainability, maintenance support performance, and, in some cases, other characteristics such as durability, safety and security.

Source: International Electrotechnical Commission Technical Committee 56 (IEC TC56): DEPENDABILITY, http://tc56.iec.ch/about/definitions.htm

What is Dependability? Another definition…

dependability: ***Trustworthiness*** of a computer system such that ***reliance*** can be ***justifiably*** placed on the service it delivers. Reliability, availability, and maintainability are aspects of dependability (adapted from Lyu, 1996).

Source: IEEE 982.1-2005

Lyu, M. R. (1996). Handbook of Software Reliability Engineering. New York: IEEE Computer Society.

(But what do ***these terms*** mean? How are they measured? 982.1 doesn't say.)

As an aside… (and in response to a discussion point Simon raised):

trust: the degree to which a user or other stakeholder has confidence that a product or system will behave as intended.

Source: ISO/IEC 25010-2011(E)

Of the Dependability Attributes…

Reliability appears to have produced the most work in the standards community, but…

Availability is the top-of-mind question for operators/maintainers, users, engineers, and acquirers:

- ***Will the software work (be operationally available to do what it is supposed to do) when I need it? Reliability is implied.***

Availability is a function of reliability. In systems engineering,

- *Reliability = MTBF, mean time between failure*
- *Availability = MTBF/(MTBF + MTTRS, mean time to restore system)*

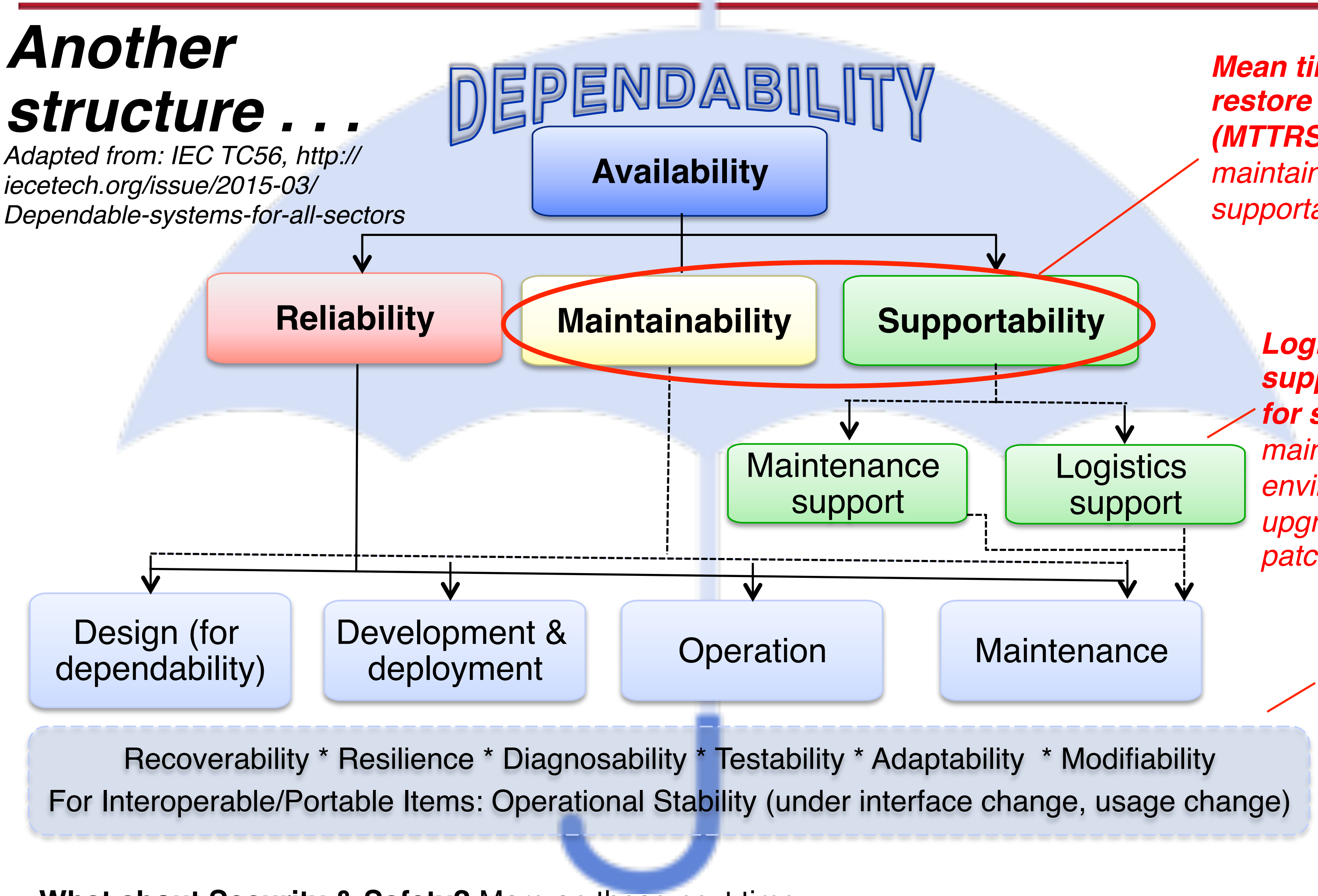
Note: This definition assumes operational systems are not taken offline for regular maintenance during a mission unless a failure occurs; some definitions equate maintenance time with downtime.

So why isn't the focus on Availability? Maybe because it's more complicated to compute:

There's a lot involved in computing MTTRS…

Another structure . . .

Adapted from: IEC TC56, http://iecetech.org/issue/2015-03/Dependable-systems-for-all-sectors



Recoverability * Resilience * Diagnosability * Testability * Adaptability * Modifiability
For Interoperable/Portable Items: Operational Stability (under interface change, usage change)

Mean time to restore system (MTTRS) *involves maintainability & supportability*

Logistics support e.g., for software: *maintenance environment, upgrades/ patches*

Some candidate sub-attributes of one or more attributes

What about Security & Safety? More on these next time.
Both support & are supported by dependability, and incorporate context-dependent attributes/rigor.

History & Work In This Area

PSM

- *The ICM* (Issue-Category-Measurable Concept-Measure) Table includes many measures applicable to dependability.*
- *PSM working groups have dealt with many topics related to dependability (security, quality, technical debt, architecture, . . .)*

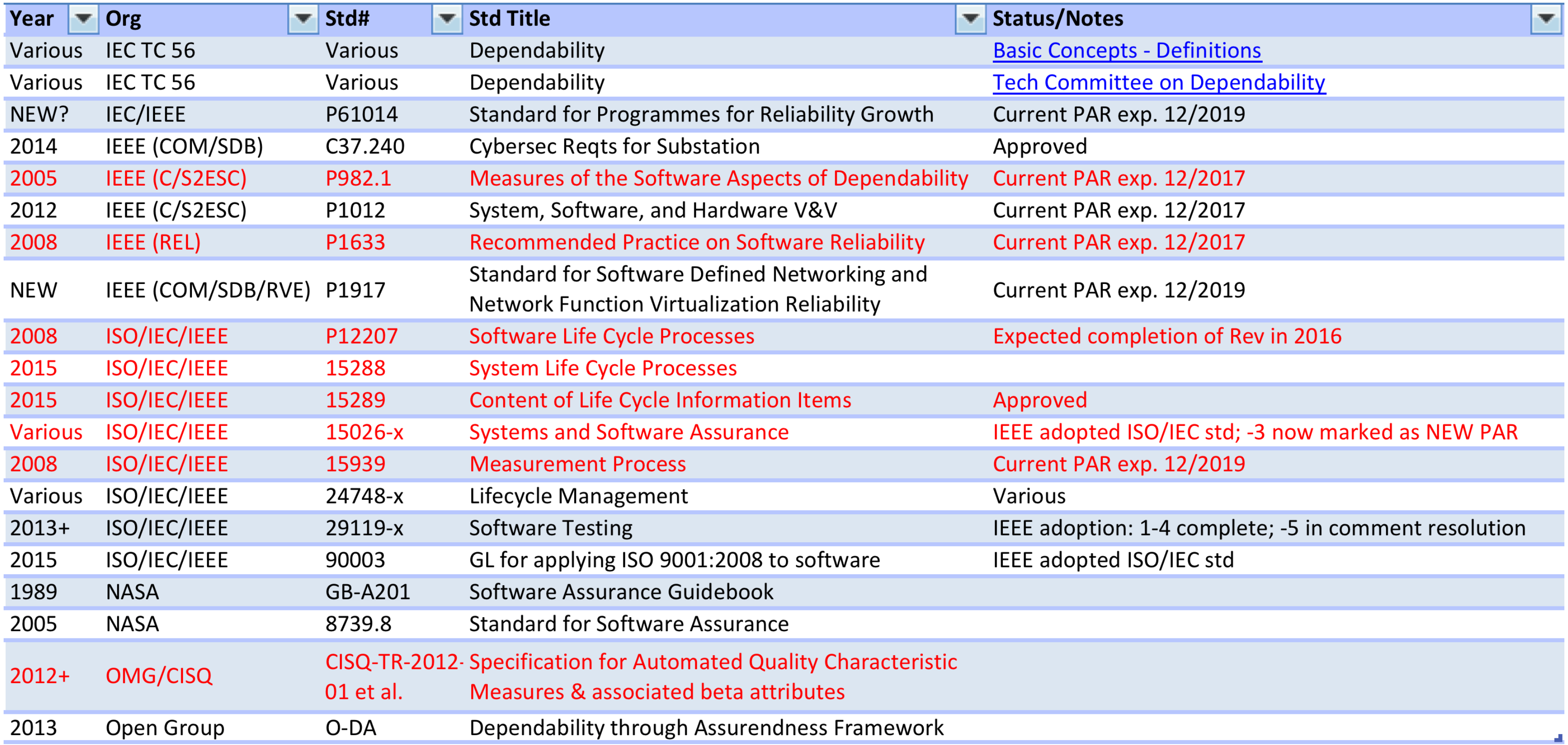
Reliability Work out of the Space Shuttle Program

Various Quality Models and Measures

- *Models: Relationships between attributes (composition/hierarchy)*
- *Measures: Data items, collection processes, and algorithms*

Other Standards…

Other Work: Related Existing/Emerging Standards



Year	Org	Std#	Std Title	Status/Notes
Various	IEC TC 56	Various	Dependability	Basic Concepts - Definitions
Various	IEC TC 56	Various	Dependability	Tech Committee on Dependability
NEW?	IEC/IEEE	P61014	Standard for Programmes for Reliability Growth	Current PAR exp. 12/2019
2014	IEEE (COM/SDB)	C37.240	Cybersec Reqts for Substation	Approved
2005	IEEE (C/S2ESC)	P982.1	Measures of the Software Aspects of Dependability	Current PAR exp. 12/2017
2012	IEEE (C/S2ESC)	P1012	System, Software, and Hardware V&V	Current PAR exp. 12/2017
2008	IEEE (REL)	P1633	Recommended Practice on Software Reliability	Current PAR exp. 12/2017
NEW	IEEE (COM/SDB/RVE)	P1917	Standard for Software Defined Networking and Network Function Virtualization Reliability	Current PAR exp. 12/2019
2008	ISO/IEC/IEEE	P12207	Software Life Cycle Processes	Expected completion of Rev in 2016
2015	ISO/IEC/IEEE	15288	System Life Cycle Processes	
2015	ISO/IEC/IEEE	15289	Content of Life Cycle Information Items	Approved
Various	ISO/IEC/IEEE	15026-x	Systems and Software Assurance	IEEE adopted ISO/IEC std; -3 now marked as NEW PAR
2008	ISO/IEC/IEEE	15939	Measurement Process	Current PAR exp. 12/2019
Various	ISO/IEC/IEEE	24748-x	Lifecycle Management	Various
2013+	ISO/IEC/IEEE	29119-x	Software Testing	IEEE adoption: 1-4 complete; -5 in comment resolution
2015	ISO/IEC/IEEE	90003	GL for applying ISO 9001:2008 to software	IEEE adopted ISO/IEC std
1989	NASA	GB-A201	Software Assurance Guidebook	
2005	NASA	8739.8	Standard for Software Assurance	
2012+	OMG/CISQ	CISQ-TR-2012-01 et al.	Specification for Automated Quality Characteristic Measures & associated beta attributes	
2013	Open Group	O-DA	Dependability through Assurendness Framework	

Red: Items of particular interest

See also PSM ICM table & measurement specifications and Software Architecture in Practice (Bass, Clements, & Kazman, 2013) Quality Attribute Trees.

Questions to Consider:

Why are you interested in measures for software aspects of dependability?

What's the role & position of the IEEE 982.1 standard today in view of

- *Related standards and guidebooks (e.g., ISO 250xx, OMG/CISQ)*
- *Applicability and usability of current set of measures in 982.1*

What questions, recommendations, challenges, & ideas do you have on

- *Growing and changing expectations about software & associated measurement gaps to be filled*
- *Significance of operational context (environment and domain)*
- *Frequency of dependability measurement*

What should we do next?

Why Are You Interested? ***Participant responses!***

I want . . .

- *to enable my group to get secure reliable software.*
- *to model modeling software reliability/maintainability effects on cost (and cost of building reliable/maintainable software).*
- *software practices/processes that produce dependable software.*
- *the **mathemagical** software number to plug into reliability equations.*
- *to work toward alignment of standards with proven practices.*
- *to learn.*
- *to help build a standard that's useful & not shelfware.*

What's the role of 982.1 today?

Most participants hadn't read the standard in detail, so we have an action to review it & discuss this question next time.

The current (2005) version

- *is based on measures that were applied to one or more high-reliability and safety-critical critical systems, e.g.,*
    - *space shuttle primary avionics software*
    - *USMC tactical systems support activity for distributed system software reliability assessment and prediction*
- *states the standard is applicable to any software system, in particular, mission critical systems.*

Question: Should 982.1 be applicable only to high-reliability, safety-critical systems

- *If yes, more such systems exist today, found in every area of our lives. Is the standard adequate, given today's environment, for these new systems?*
- *If not, is the standard appropriate/usable for other relevant domains?*

Increasing Need to Evaluate SW Aspects of Dependability

Software State of Affairs	Comments
Ubiquity *(esp. of high-reliability/safety critical software-reliant systems)*	Software's in just about everything and is becoming more autonomous.
Interconnections	Software is hyperconnected and these connections change
Provenance complex/difficult to ascertain *(supply chain assurance)*	COTS, OSS, reuse
Interdependencies	Technical & organizational
Volatility	Frequent software changes; changes in interfacing software systems
Increased attack surfaces	More software, more interfaces, more opportunities for adversaries to penetrate
Emergent behavior displayed	Plug 'n' play has unpredictable outcomes

Current and Emerging Needs

(mix of measurement process & product needs)

Needs Related to Measures Software Aspects of Dependability
Elevator speech to explain why dependability is important (so we can build it in and measure it)
Dependability measures to support decision-making and planning (e.g., early indication of product issues that could lead to dependability problems)
Context-appropriate measures (context: environment + domain) & examples applicable to specific domains
Measures for dependability-related architecturally-significant attributes
Architecture & process patterns that lead to dependability (measures to help identify)
Increased understanding of valid measurement practice (e.g., understanding of scales; 15939, PSM)
More frequent dependability-related checkpoints (V&V-like activities)
Dependability measurement for systems at scale (e.g., ultra-large-scale systems)
Measures that help improve dependability in a standard that enables success!

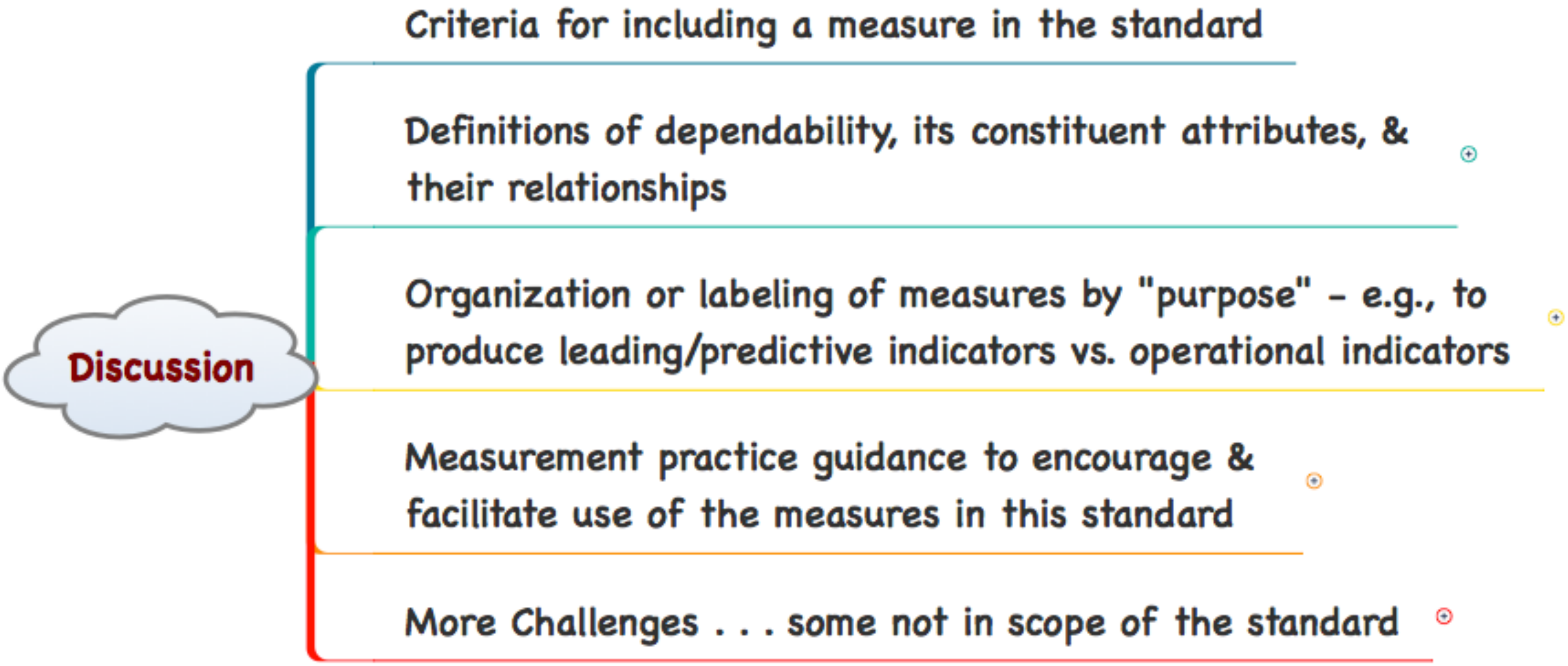
Back to the Standard & Its Content

How can we select, define, & explain the set of software attributes that characterize dependability?

- *How can we measure these attributes throughout the lifecycle?*
- *Which should be implemented broadly? Which can we affordably implement today?*

How can IEEE P982.1 and PSM advance dependability measurement practice?

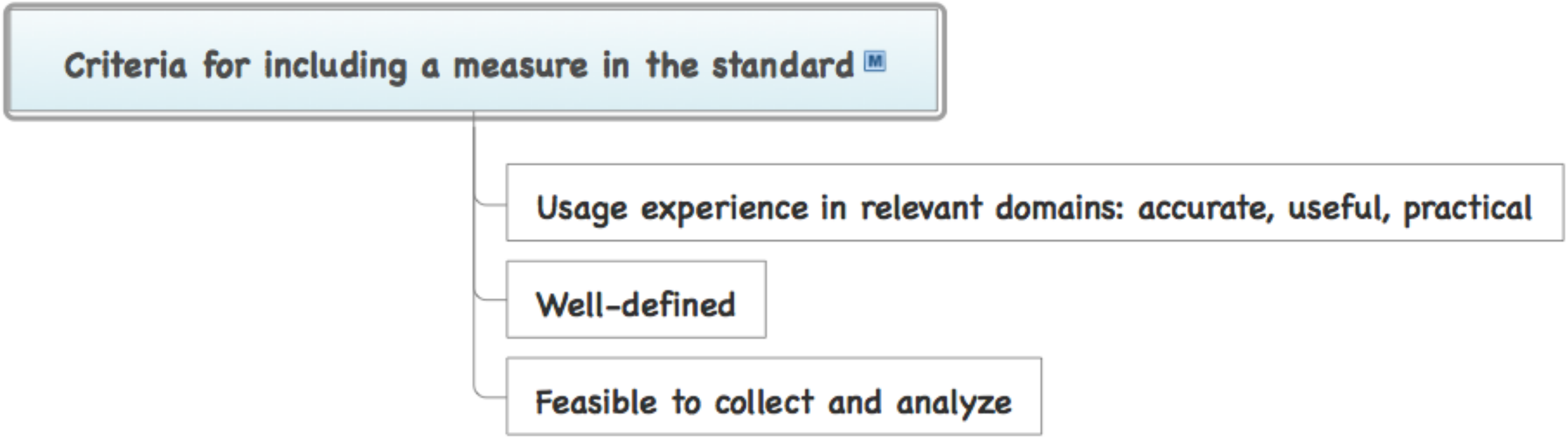
Discussion of 982.1 Revision Needs: Topics



Discussion

- Criteria for including a measure in the standard
- Definitions of dependability, its constituent attributes, & their relationships
- Organization or labeling of measures by "purpose" – e.g., to produce leading/predictive indicators vs. operational indicators
- Measurement practice guidance to encourage & facilitate use of the measures in this standard
- More Challenges . . . some not in scope of the standard

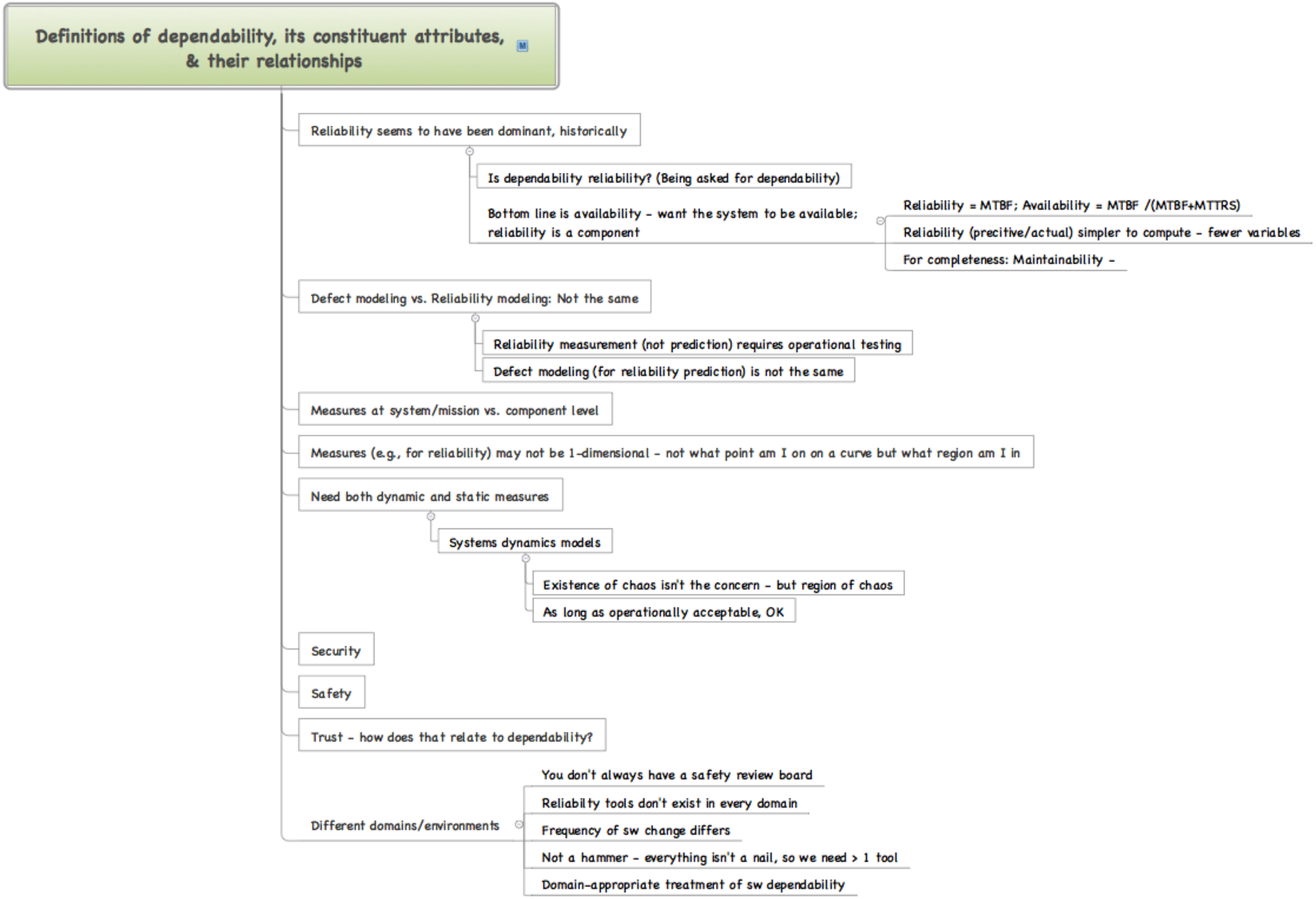
Discussion Capture: Criteria for including a measure . . .



Criteria for including a measure in the standard

- Usage experience in relevant domains: accurate, useful, practical
- Well-defined
- Feasible to collect and analyze

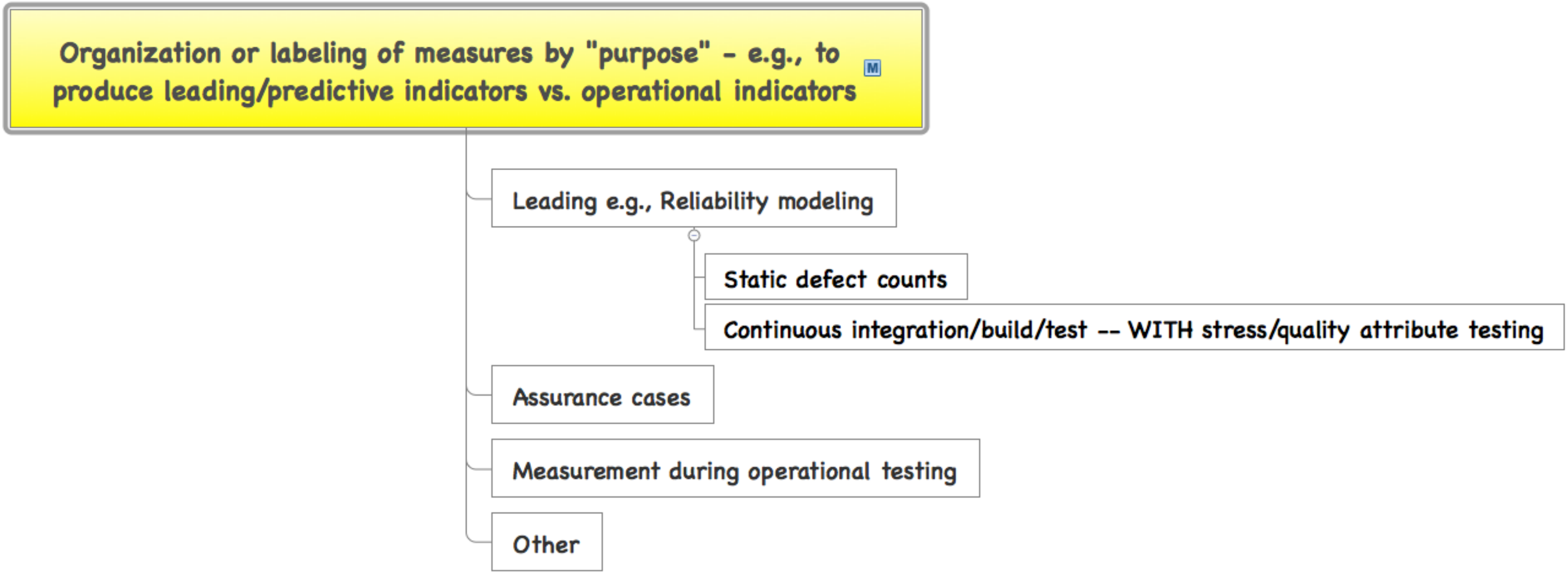
Discussion Capture: Definitions . . .



Definitions of dependability, its constituent attributes, & their relationships

- Reliability seems to have been dominant, historically
  - Is dependability reliability? (Being asked for dependability)
  - Bottom line is availability – want the system to be available; reliability is a component
    - Reliability = MTBF; Availability = MTBF /(MTBF+MTTRS)
    - Reliability (precitive/actual) simpler to compute – fewer variables
    - For completeness: Maintainability –
- Defect modeling vs. Reliability modeling: Not the same
  - Reliability measurement (not prediction) requires operational testing
  - Defect modeling (for reliability prediction) is not the same
- Measures at system/mission vs. component level
- Measures (e.g., for reliability) may not be 1-dimensional – not what point am I on on a curve but what region am I in
- Need both dynamic and static measures
  - Systems dynamics models
    - Existence of chaos isn't the concern – but region of chaos
    - As long as operationally acceptable, OK
- Security
- Safety
- Trust – how does that relate to dependability?
- Different domains/environments
  - You don't always have a safety review board
  - Reliabilty tools don't exist in every domain
  - Frequency of sw change differs
  - Not a hammer – everything isn't a nail, so we need > 1 tool
  - Domain-appropriate treatment of sw dependability

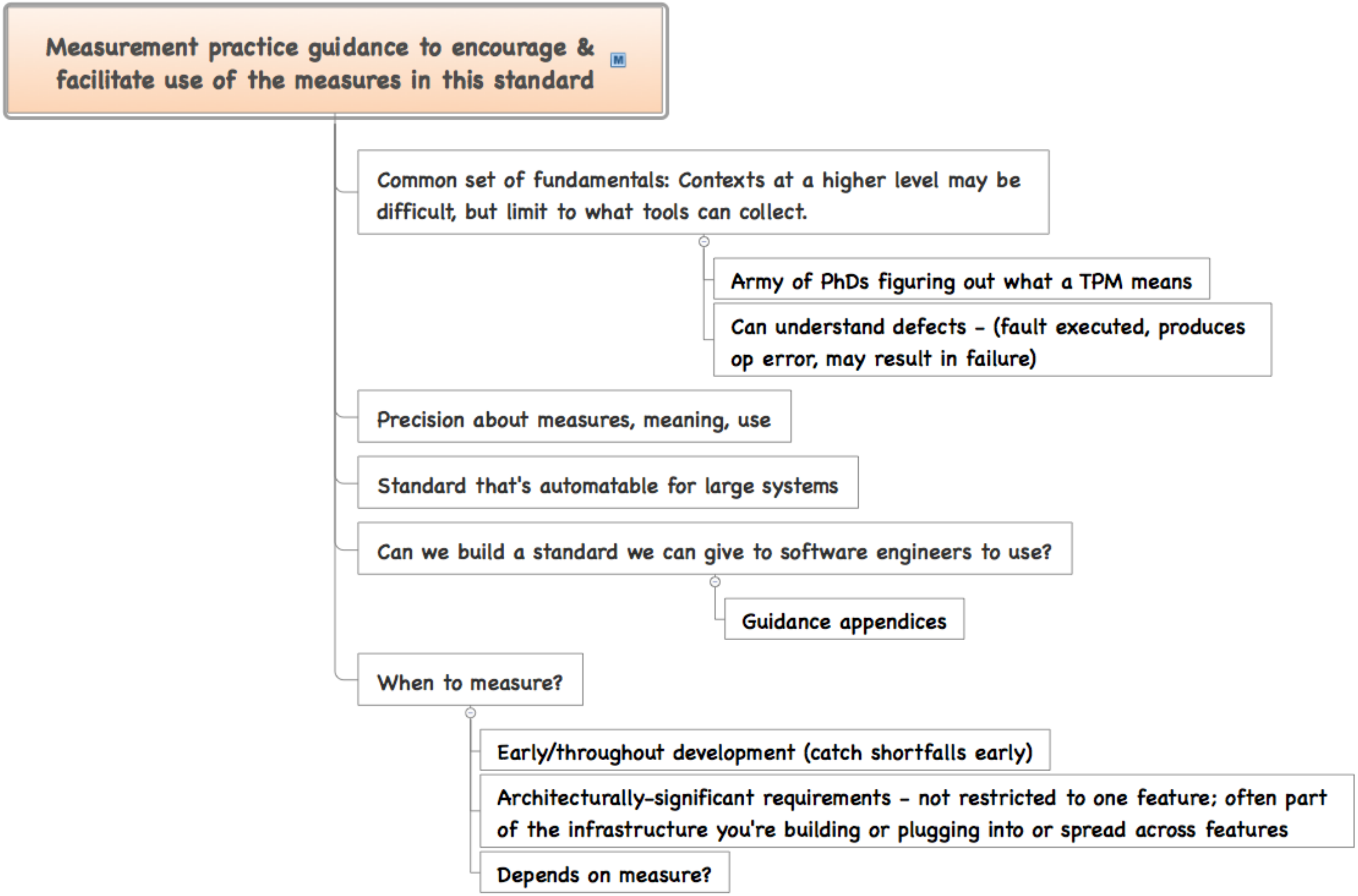
Discussion Capture: Organization . . .



Organization or labeling of measures by "purpose" – e.g., to produce leading/predictive indicators vs. operational indicators

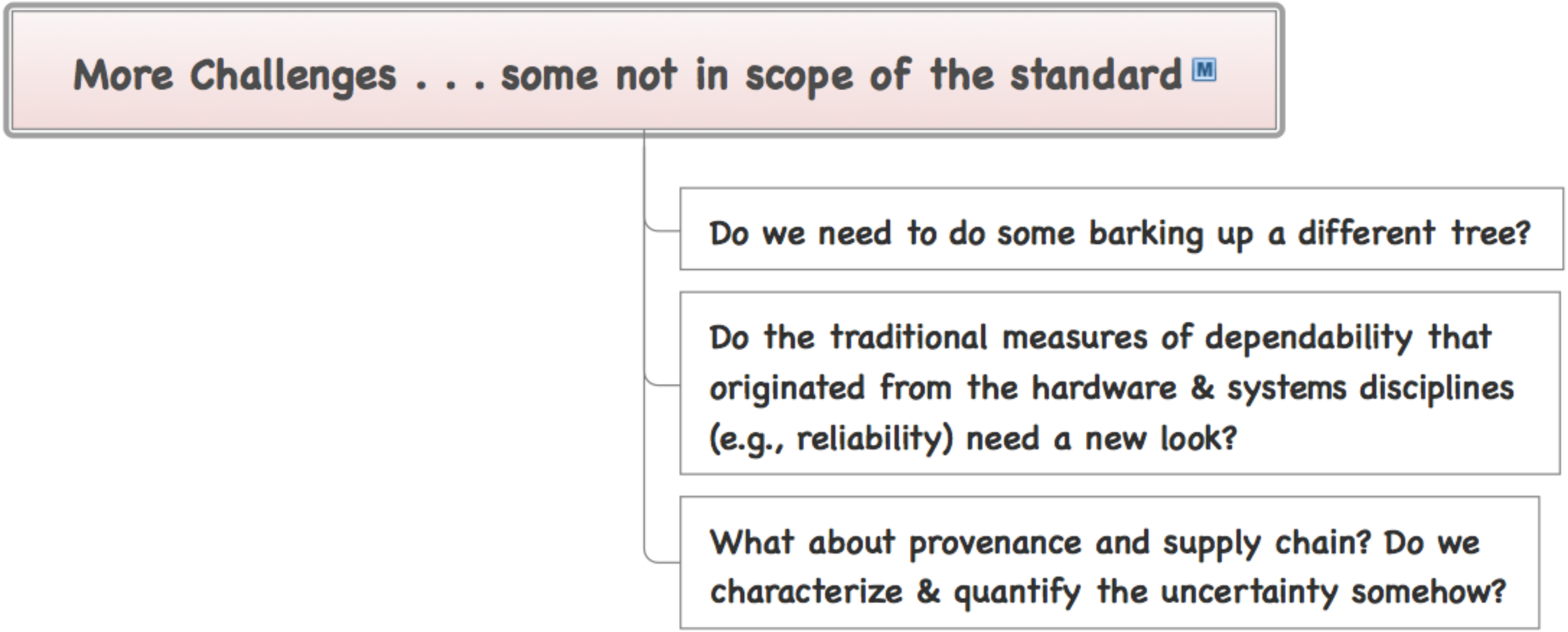
- Leading e.g., Reliability modeling
  - Static defect counts
  - Continuous integration/build/test -- WITH stress/quality attribute testing
- Assurance cases
- Measurement during operational testing
- Other

Discussion Capture: Measurement Practice . . .



- Measurement practice guidance to encourage & facilitate use of the measures in this standard
  - Common set of fundamentals: Contexts at a higher level may be difficult, but limit to what tools can collect.
    - Army of PhDs figuring out what a TPM means
    - Can understand defects – (fault executed, produces op error, may result in failure)
  - Precision about measures, meaning, use
  - Standard that's automatable for large systems
  - Can we build a standard we can give to software engineers to use?
    - Guidance appendices
  - When to measure?
    - Early/throughout development (catch shortfalls early)
    - Architecturally-significant requirements – not restricted to one feature; often part of the infrastructure you're building or plugging into or spread across features
    - Depends on measure?

Discussion Capture: More Challenges . . .



More Challenges . . . some not in scope of the standard

- Do we need to do some barking up a different tree?
- Do the traditional measures of dependability that originated from the hardware & systems disciplines (e.g., reliability) need a new look?
- What about provenance and supply chain? Do we characterize & quantify the uncertainty somehow?

Some Important Questions Raised…

Ask: Who pays for defects? Failures?

- *What gets measured, gets attention*
- *Stakeholder alignment and motivation*

Ask: Are the culture and management focus conducive to quality?

Do we need a standard for a software dependability engineering culture ☺?

Conclusions

The workshop discussion was illuminating!

Many more organizations & programs can and should use a proven set of dependability attribute measures than are currently doing so.

We don’t have consistent definitions for dependability, despite the existence of standards for dependability and its attributes/aspects, especially for software.

There’s active work in both practical methods/tools and research, but significant gaps remain.

Recommendations

Look at the software dependability measurement problem from a different perspective:

- *Can we fit the square software peg into the round, traditional RMA (reliability-maintainability-availability) hole? Maybe, but*
  - *Software doesn't "wear out" like hardware.*
  - *Measuring whether it will grow or degrade in dependability over its lifetime--and to what extent—requires different methods than those used for hardware.*

Continue the PSM effort as a collaboration with the P982.1 WG and share results.

- *Support revision of the standard.*
- *Find "homes" for out-of-scope topics, i.e., PSM white papers and other publications.*

Recommended Homework

Determine the positioning of 982.1 with respect to related standards & practices:

- *Review related standards*
  - *Examine quality/dependability models, measures, and measure definitions.*
  - *What gaps or issues are evident in these standards? Are they similar to the ones we captured for 982.1?*
  - *What do these standards include that 982.1 does not?*
- *WG Chair will review, characterize, and depict the landscape.*

In terms of applicability / usability of 982.1:

- *Look for case studies; consider a survey of some sort.*
- *Check proceedings of IEEE/IFIP International Conference on Dependable Systems and Networks (DSN**).***
- *Answer (cite or if experience based, so indicate): If 982.1 has not been used, why not? How do organizations concerned about dependability measure and manage it today?*

Next Steps/Actions—PSM

- *Provide our PSM workshop's feedback to the IEEE WG.*
- *Determine next activity for PSM—perhaps gather at the next meeting of the SW Cost IPT, or hold a Skype session of some sort.*
- *Workshop participants will receive a note asking if they want to join the WG (get the standard for free to review).*

Continuing IEEE P982.1 WG Activity

- *Status/progress of P982.1 WG will be posted on the Public wiki, https://ieee-sa.imeetcentral.com/p9821wgpublic/*
- *Updates will be provided at subsequent PSM events – with a follow-on workshop, if there's interest*
- *We'll pursue opportunities for continued engagement with PSM.*
- *If you'd like to join the WG or stay informed about P982.1's progress, please send me email at rc@sei.cmu.edu*

Workshop #5: Measuring Software Aspects of Dependability: Revisiting IEEE 982.1

Additional Materials: Workshop Aids

Introductions

Flip chart capture

Who are you (name & whatever else you'd like us to know)?

What brought you to PSM?

Why is the topic of software aspects of dependability important to you?

What would you like to have happen as a result of our work today?

Action Item Log: Workshop # 5, Measuring SW Aspects of Dependability

Item #	Due	Assignee	Description

Idea Log: Workshop # 5, Measuring SW Aspects of Dependability

Item #	Initiator	Description

Workshop Evaluation

At the start, you told us

- *Why the topic of software aspects of dependability is important to you*
- *What you would like to have happen as a result of our work today*

What was helpful (or unhelpful)?

- **+**
- **-**

What should we do that's different next time?

- ***Δ***