Generating Supportive Hypotheses in Introducing Formal Methods using a Software Process Improvement Model

<u>S. KUSAKABE,</u> H. LIN, Y. OMORI, and K. ARAKI

Grad. Sch. of Info. Sci. & Elec. Eng.,

Kyushu University

Outline

- Introduction
- Process Improvement Model : CMMI-DEV
- Generating Hypotheses through the Model
- Examining Hypotheses: PSP with VDM
- Concluding Remarks

Outline

- Introduction
- Process Improvement Model : CMMI-DEV
- Generating Hypotheses through the Model
- Examining Hypotheses: PSP with VDM
- Concluding Remarks

Background: Process model can help us?

Using formal methods (FM) is a promising approach to

- reliable and dependable systems
  - recommended in standards: ISO/IEC15408, IEC61508
- high quality products within shorter period

Concern: When, where, how to introduce FM into development process in an effective and feasible manner

"Effective **processes** provide **a vehicle for introducing and using new technology** in a way that best meets the business objectives of the organization." (from CMMI)

Background: Formal methods can help us?

Using FM is a promising approach to

- developing reliable and dependable systems
  - recommended in standards: ISO/IEC15408, IEC61508
- developing high quality products within shorter period

Another view point: These issues are also important in process improvement. Using FM is also an promising approach to software process improvement ?

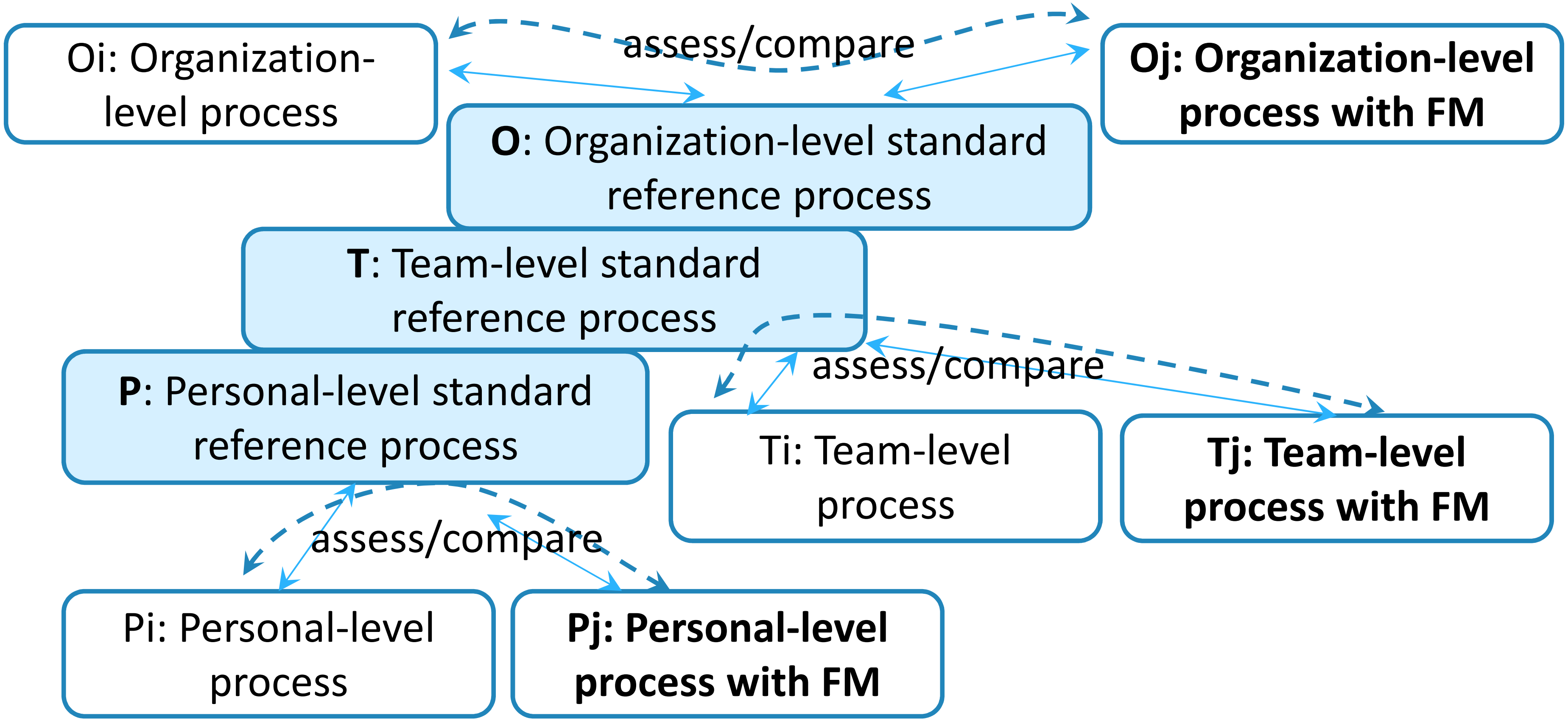
FM & SE can benefit each other?

We consider these issues through standard process models.

(Representation of best-practices)

Standard Process & Process Instances w/wo Formal Methods

Framework: useful both in introducing FM and in improving process while actual processes may be different from each other.



Expectation

Each process instance: a derived version of the standard process model, and we can share the lessons learned by using the standard process model as a reference.

- We can effectively introduce FM into a well-defined and analyzable software process.
- We can effectively improve well-defined & analyzable process with FM by using process improvement model.
  - well-defined -> easy to introduce FM strategically
  - analyzable -> easy to compare/assess process (conduct PDCA cycle) in tailoring process with FM

Outline

- Introduction
- Process Improvement Model : CMMI-DEV
- Generating Hypotheses through the Model
- Examining Hypotheses: PSP with VDM
- Concluding Remarks

Process Improvement Models

Process reference models / assessment models

- **CMMI (Capability Maturity Model Integration)***
- ISO/IEC 15504 (SPICE: Software Process Improvement & Capability dEtermination)
  - Automotive SPICE
  - SPICE for SPACE
  - Others
    - Banking SPICE
    - Medi SPICE
    - Enterprise SPICE, ...

There exit process implementation templates for developers:
TSP(Team Software Process)*
PSP(Personal Software Process)*

* Service marks and registered marks of Carnegie Mellon University Software Engineering Institute.

CMMI-DEV

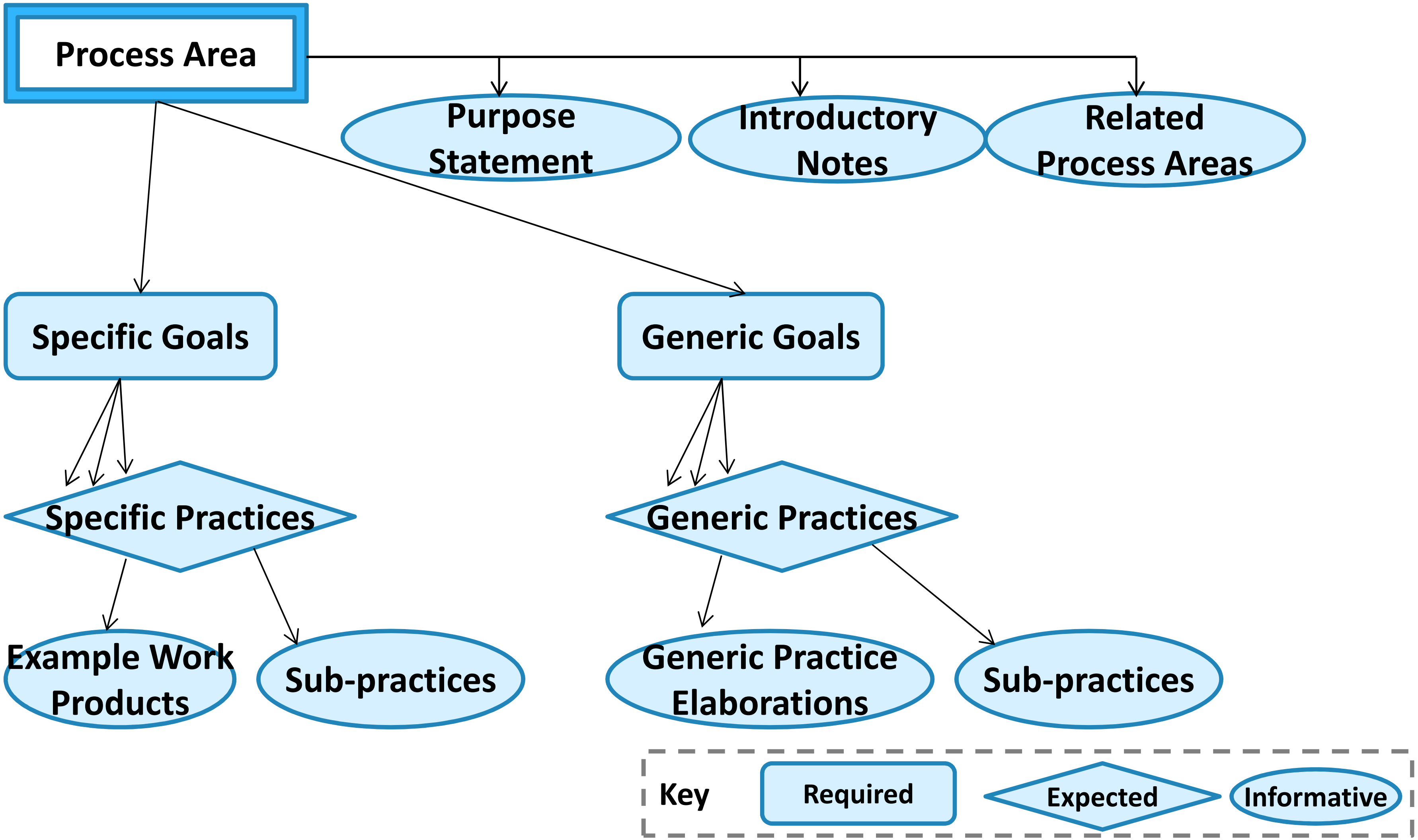
- CMMI Framework
  - Process model for Acquisition, Service, **<u>Development</u>**
  - 16 core process areas plus specific areas
    - each process area contains a set of goals
      - generic goal : characteristics that must be present to institutionalize processes that implement a process area.
      - specific goal : unique characteristics that must be present to satisfy the process area.
- Two assessment/improvement paths
  - staged representation: maturity level
    - A maturity level consists of related specific and generic practices for a predefined set of process areas that improve the organization's overall performance.
  - continuous representation: capability level
    - concerned with selecting both a particular process area to improve and the desired capability level for that process area.
    - four process area categories: Process Management, Project Management, Engineering, and Support

Process Areas (staged/continuous)

Maturity level	Process area name	Category
2	Requirements Management (REQM)	Project Management
2	Project Planning (PP)	Project Management
2	Project Monitoring and Control (PMC)	Project Management
2	Supplier Agreement Management (SAM)	Project Management
2	Measurement and Analysis (MA)	Support
2	Process and Product Quality Assurance (PPQA)	Support
2	Configuration Management (CM)	Support
3	Requirements Development (RD)	Engineering
3	Technical Solution (TS)	Engineering
3	Product Integration (PI)	Engineering
3	Verification (VER)	Engineering
3	Validation (VAL)	Engineering
3	Organizational Process Focus (OPF)	Process Management
3	Organizational Process Definition (OPD)	Process Management
3	Organizational Training (OT)	Process Management
3	Integrated Project Management (IPM)	Project Management
3	Risk Management (RSKM)	Project Management
3	Decision Analysis and Resolution (DAR)	Support
4	Organizational Process Performance(OPP)	Process Management
4	Quantitative Project Management (QPM)	Project Management
5	Organizational Performance Management (OPM)	Process Management
5	Causal Analysis and Resolution (CAR)	Support

Continuous representation allows various combinations → tailoring to formal methods

Model Components and Structure

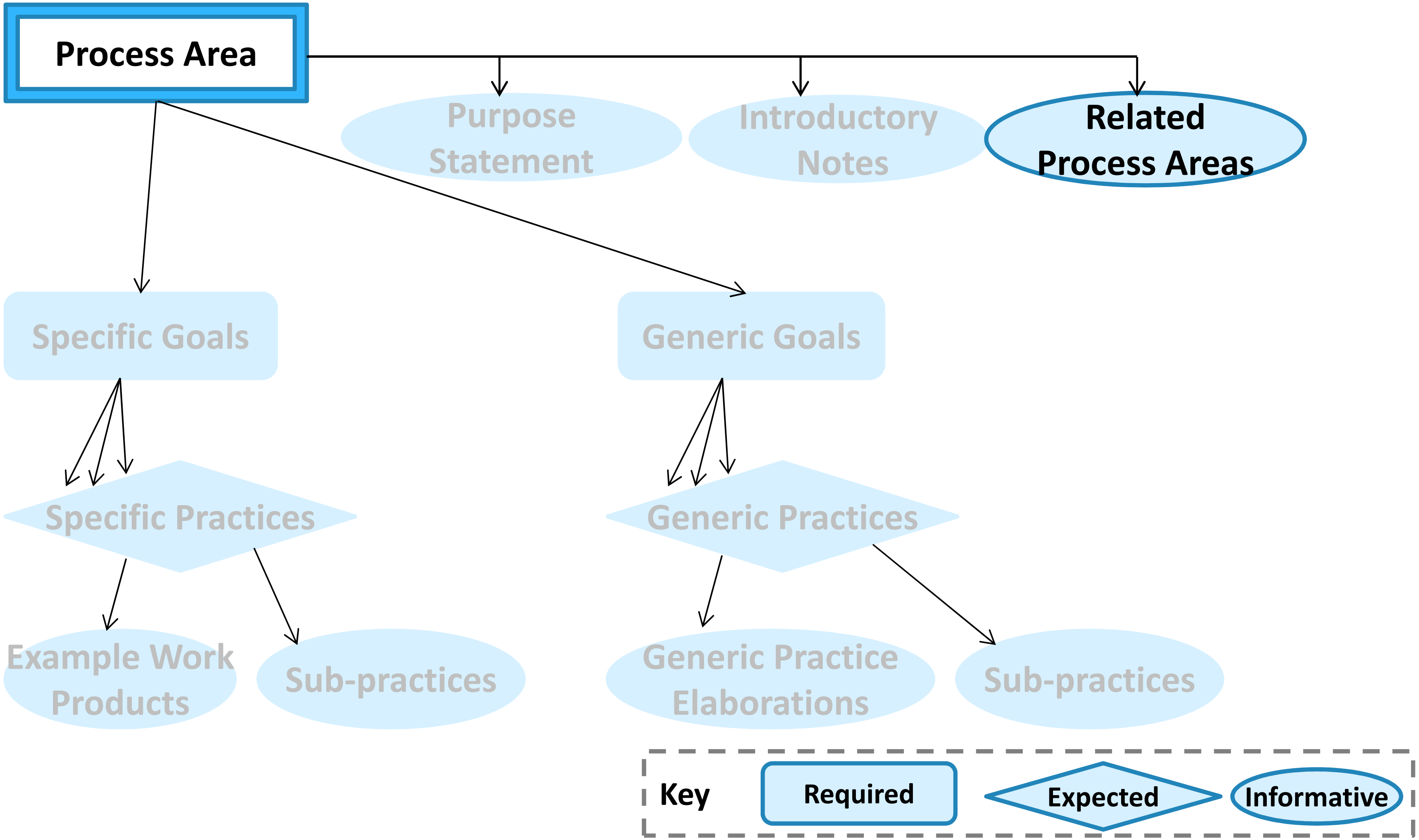


The Number of PA-specific Components

Maturity Level	Process Area	Specific Goal	Specific Practice	Subpractice	Work Products (detailed)	Work Products (high-level)
1	0	0	0	0	0	0
2	7	15	54	221	135	15
3	11	26	86	404	227	27
4	2	3	11	61	27	4
5	2	5	14	69	31	4
Total	22	49	165	755	420	50

hundreds of elements in details

Focused Component: Related Process Areas



Outline

- Introduction
- Process Improvement Model : CMMI-DEV
- Generating Hypotheses through the Model
- Examining Hypotheses: PSP with VDM
- Concluding Remarks

Related Process Areas (in table)

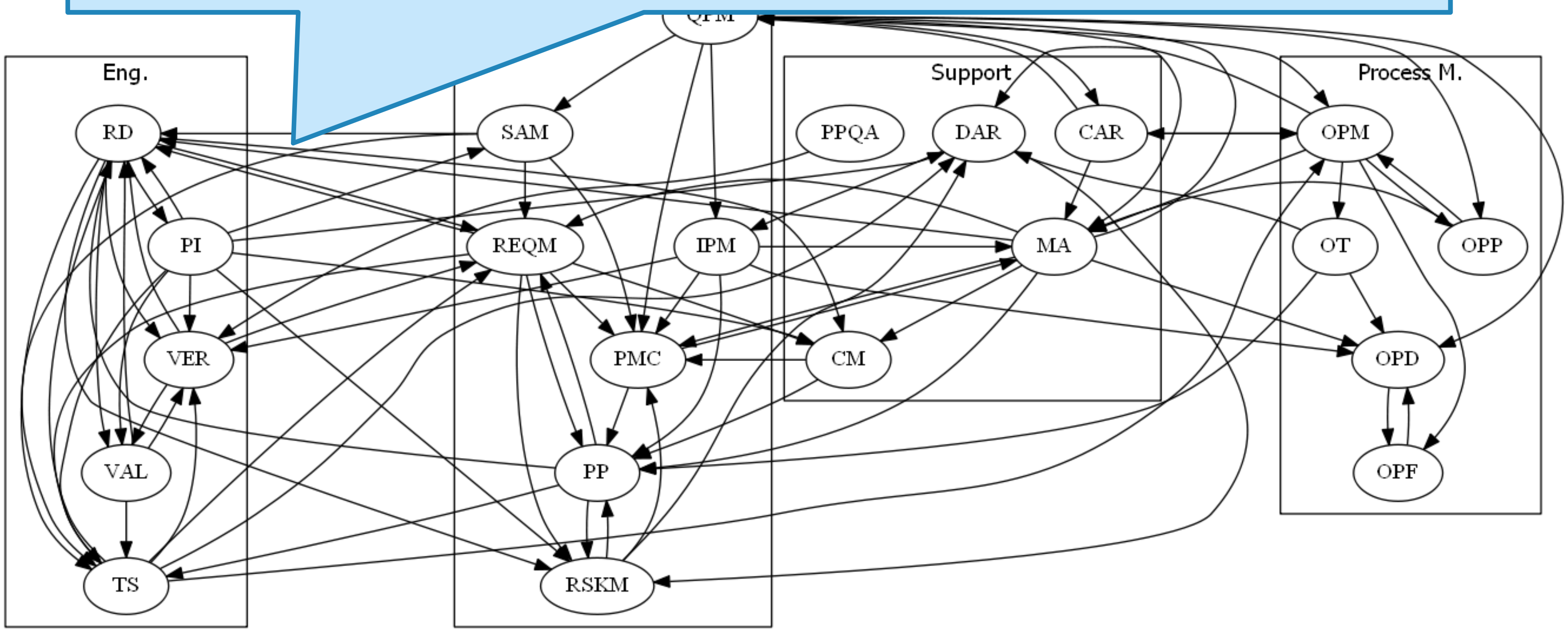
refer to P.A.	REQM	PP	PMC	SAM	MA	PPQA	CM	RD	TS	PI	VER	VAL	OPF	OPD	OT	IPM	RSKM	DAR	OPP	QPM	OPM	CAR
REQM		○	○				○	○	○								○					
PP	○				○			○	○								○					
PMC		○			○																	
SAM	○		○					○	○													
MA	○	○	○				○	○						○						○		
PPQA											○											
CM		○	○																			
RD	○						○		○	○	○	○					○					
TS	○							○			○							○			○	
PI				○			○	○	○		○	○					○	○				
VER	○							○				○										
VAL								○	○		○											
OPF														○								
OPD													○									
OT		○												○				○				
IPM		○	○		○						○			○								
RSKM		○	○															○				
DAR																○	○					
OPP					○																○	
QPM			○	○	○									○		○			○		○	○
OPM					○								○		○			○	○			○
CAR					○															○	○	

ML2 ML3 ML4 ML5

We illustrate and analyze the dependencies among process areas in a bird's-eye view by using tools such as Graphviz and Gephi.

Dependency Graph of Related Process Areas (Four Categories)

For example, effects of improving processes in Engineering category may propagate to processes in other categories.



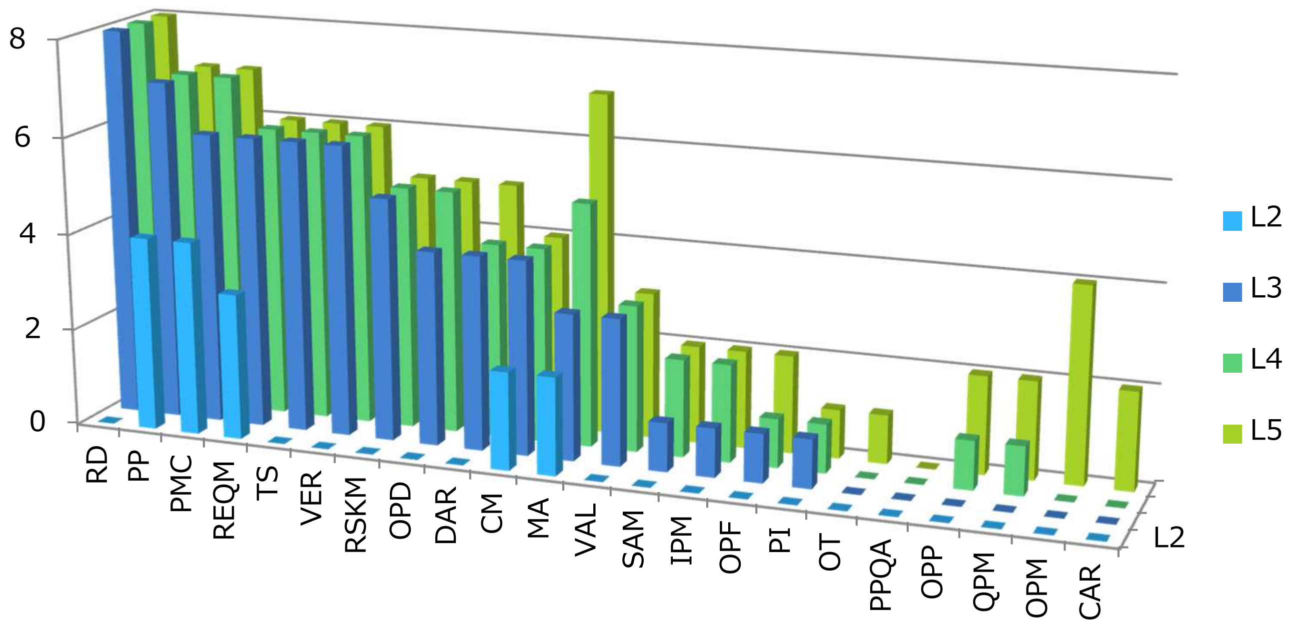
Investigating Indirect Effects

Meta-level Network Analysis (Network Theory)

- **Degree centrality:**
  - "*An important node is involved in large number of interactions*"
- **Betweenness centrality:**
  - "*An important node will lie on a high proportion of paths between other nodes in the network.*"

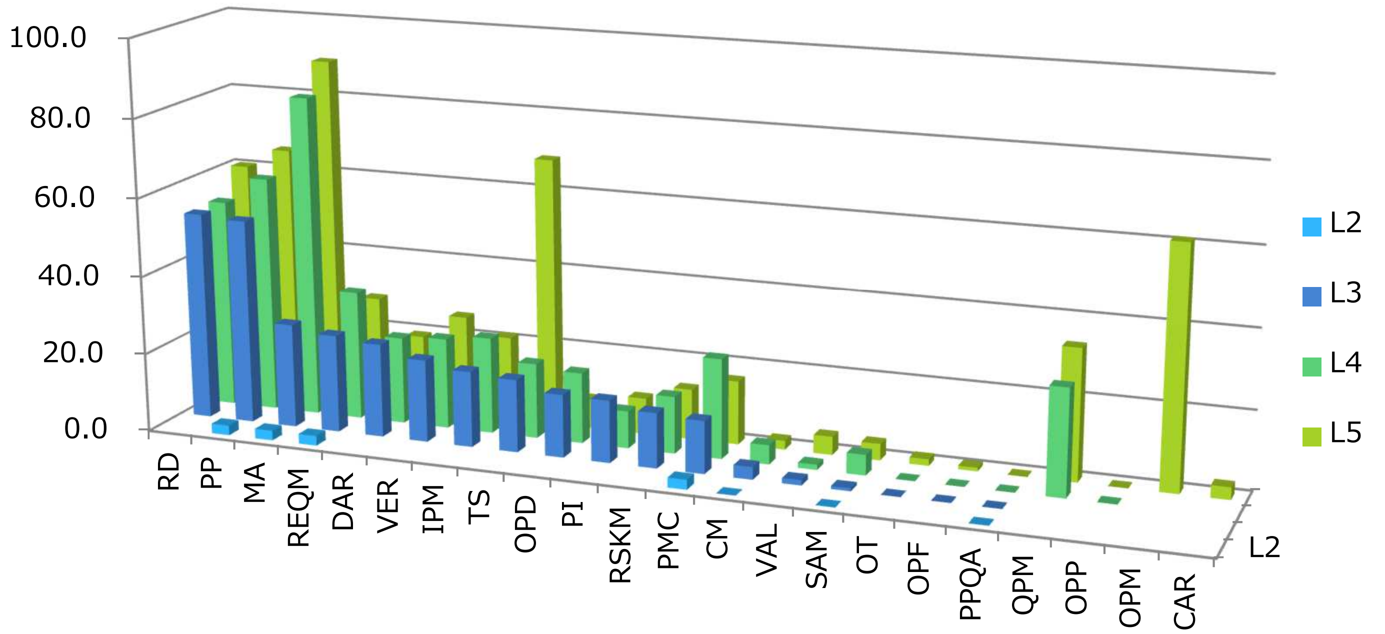
In-Degree centrality (Sorted by L3-value)

Id	L2	L3	L4	L5
RD		8	8	8
PP	4	7	7	7
PMC	4	6	7	7
REQM	3	6	6	6
TS		6	6	6
VER		6	6	6
RSKM		5	5	5
OPD		4	5	5
DAR		4	4	5
CM	2	4	4	4
MA	2	3	5	7
VAL		3	3	3
SAM	0	1	2	2
IPM		1	2	2
OPF		1	1	2
PI		1	1	1
OT		0	0	1
PPQA	0	0	0	0
OPP			1	2
QPM			1	2
OPM				4
CAR				2



Betweenness centrality (Sorted by L3-value)

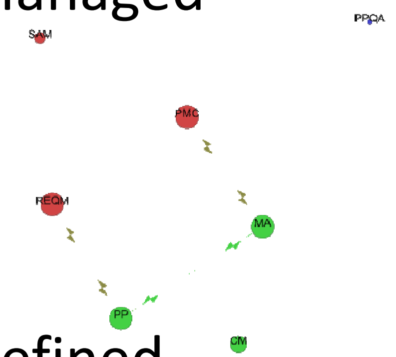
	L2	L3	L4	L5
RD		53.1	53.5	60.3
PP	2.5	52.4	60.5	65.4
MA	2.5	26.5	82.2	89.3
REQM	2.5	24.9	32.9	28.2
DAR		23.8	22.2	19.3
VER		21.0	23.1	25.6
IPM		19.3	24.5	21.3
TS		18.5	19.1	68.5
OPD		16.0	18.0	7.4
PI		15.8	9.5	9.2
RSKM		14.0	14.5	12.8
PMC	2.5	13.4	25.3	16.2
CM	0.0	3.2	4.9	2.3
VAL		1.3	1.3	4.7
SAM	0.0	0.8	5.2	4.2
OT		0.0	0.0	1.7
OPF		0.0	0.0	0.9
PPQA	0.0	0.0	0.0	0.0
QPM			27.2	33.4
OPP			0.0	0.0
OPM				61.2
CAR				3.1



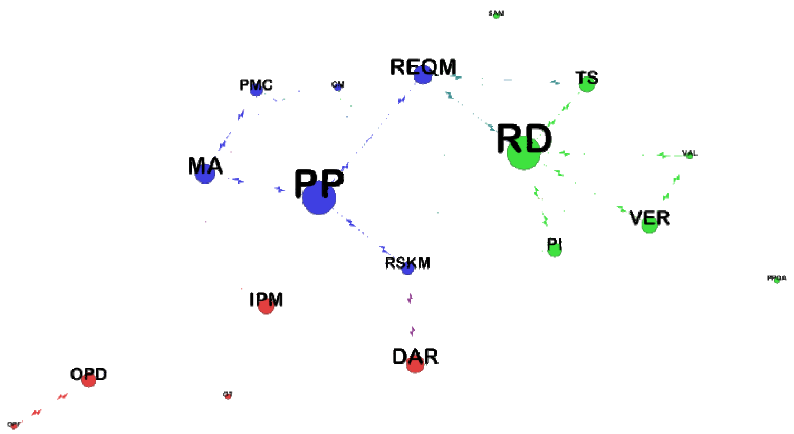
Going up Maturity Levels

RD & PP become important from ML3 and some other PAs in ML4/5.

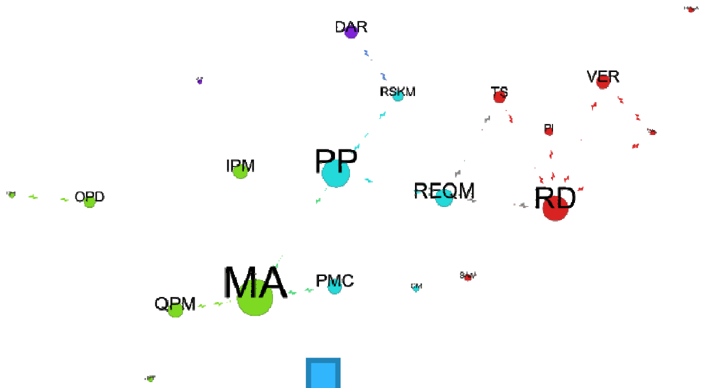
- ML1: Initial
- ML2: Managed



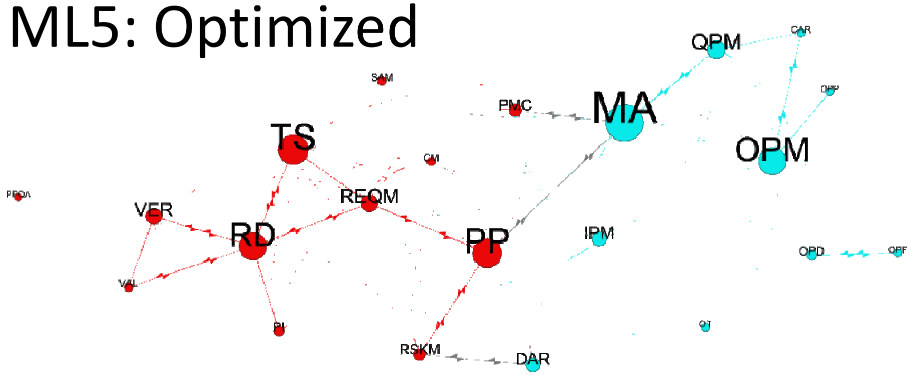
- ML3: Defined



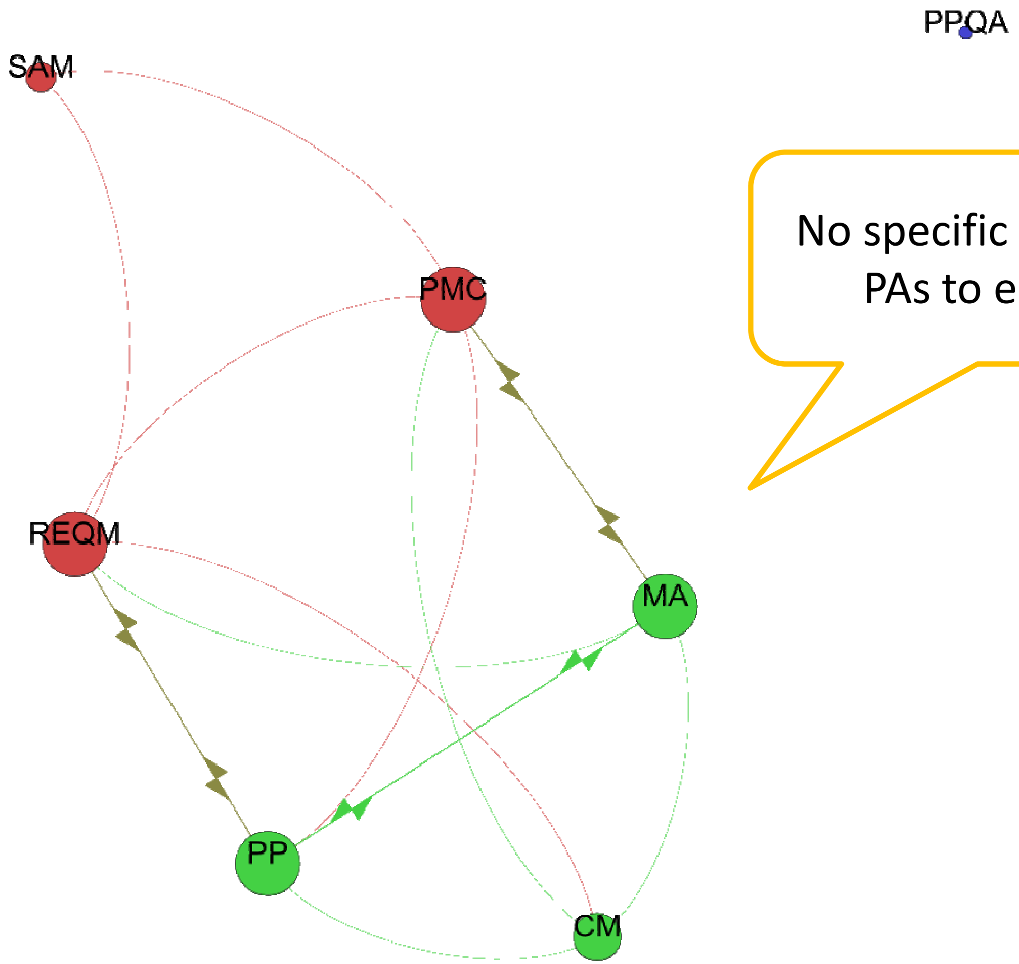
- ML4: Quantitatively Managed



- ML5: Optimized

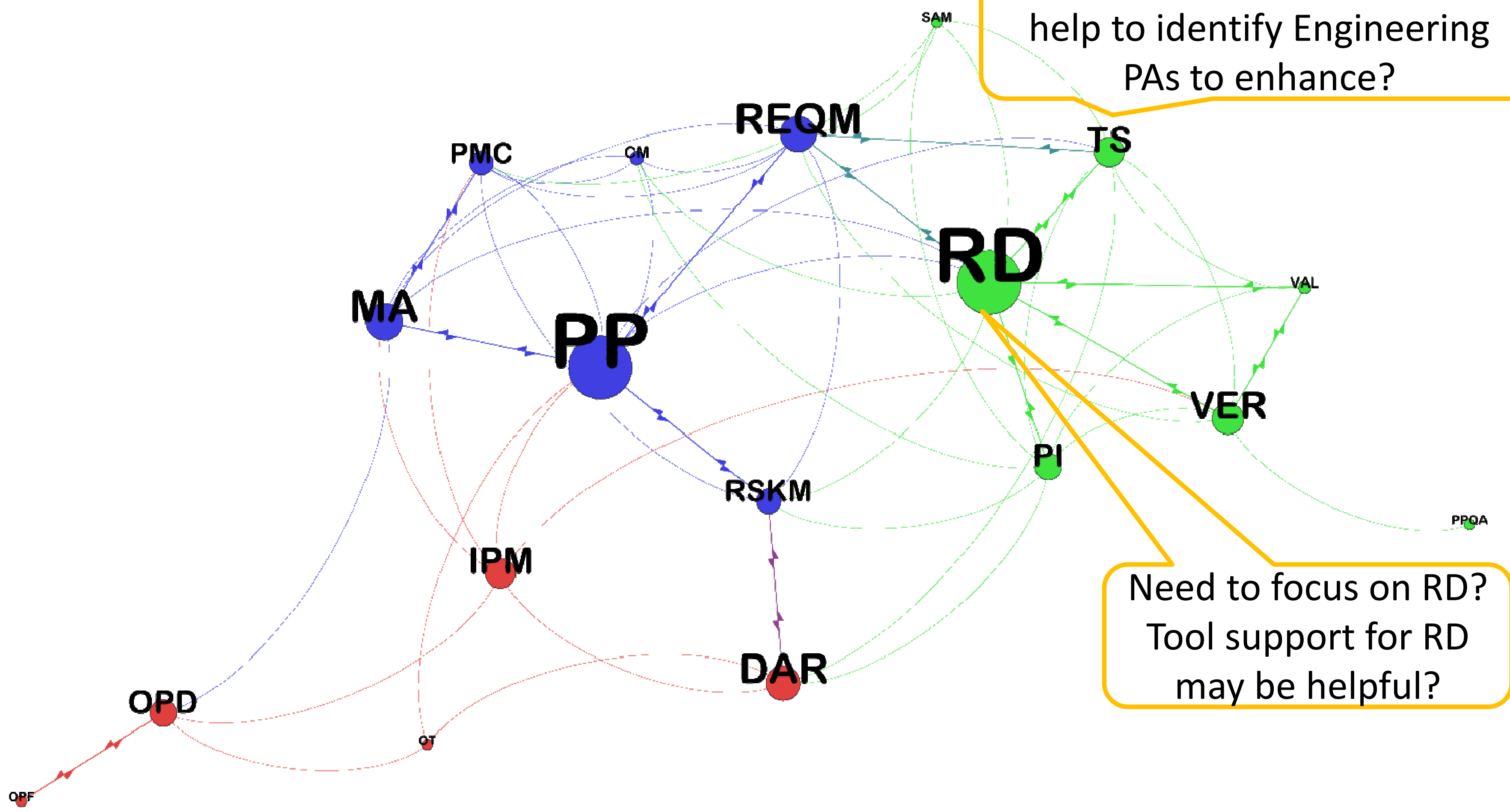


ML 2 (Managed)
- betweenness centrality -



No specific Engineering PAs to enhance?

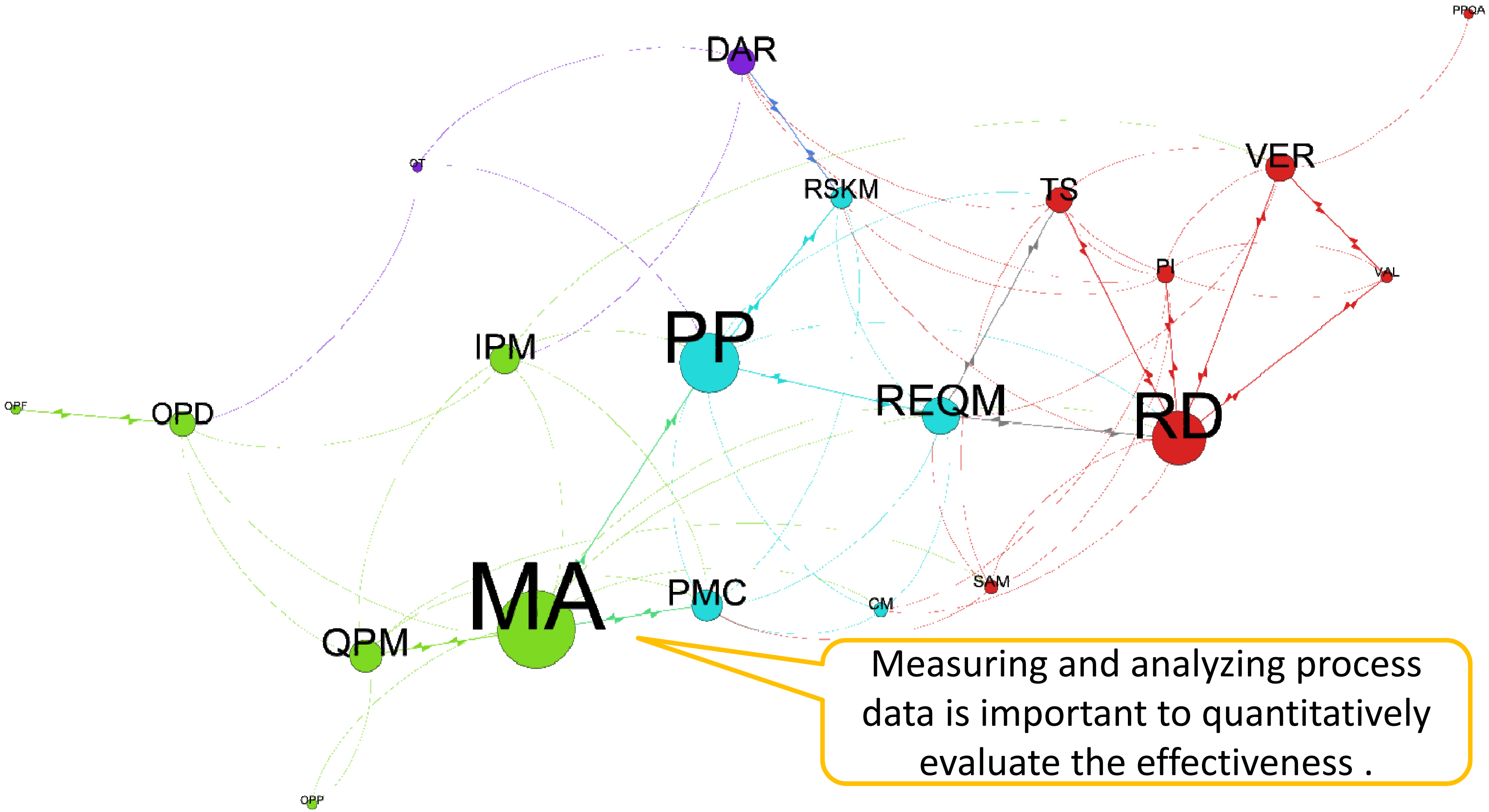
ML 3 (Defined)
- betweenness centrality -



Having defined process may help to identify Engineering PAs to enhance?

Need to focus on RD? Tool support for RD may be helpful?

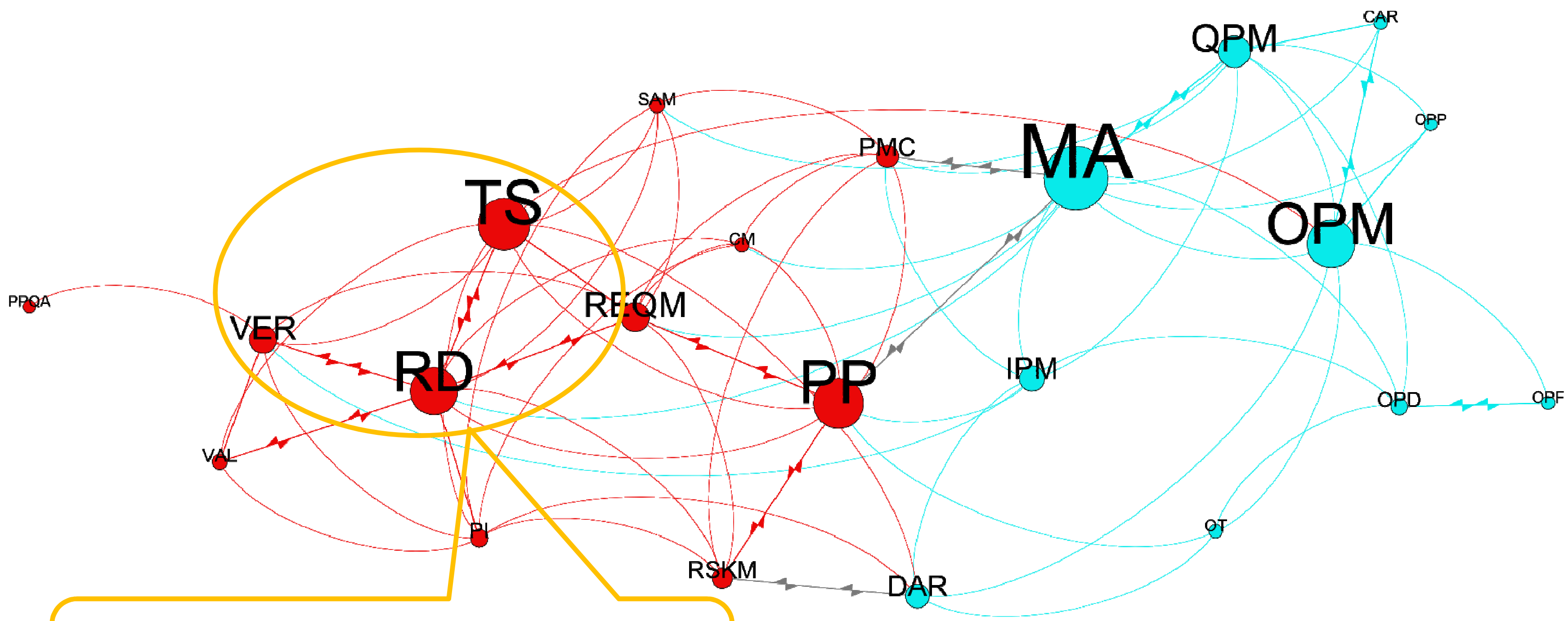
ML 4 (Quantitatively Managed)
- betweenness centrality -



Measuring and analyzing process data is important to quantitatively evaluate the effectiveness .

ML 5 (Optimizing)

- betweenness centrality -



There is a set of Engineering PAs to continuously optimize the process?

Candidate Hypotheses

- ML2: No specific Engineering PAs to enhance.

-> Anyone who cannot make a decision are at this level?

- ML3: Use FM effective in RD, Tool to support RD may be helpful.

(-> tools being developed in this project.)

- ML3: Having defined process may help to identify Engineering PAs to enhance.
- ML4: Measuring and analyzing process data is important to quantitatively evaluate the effectiveness.

-> ongoing with measurable /analyzable process such as TSP(TSPi)

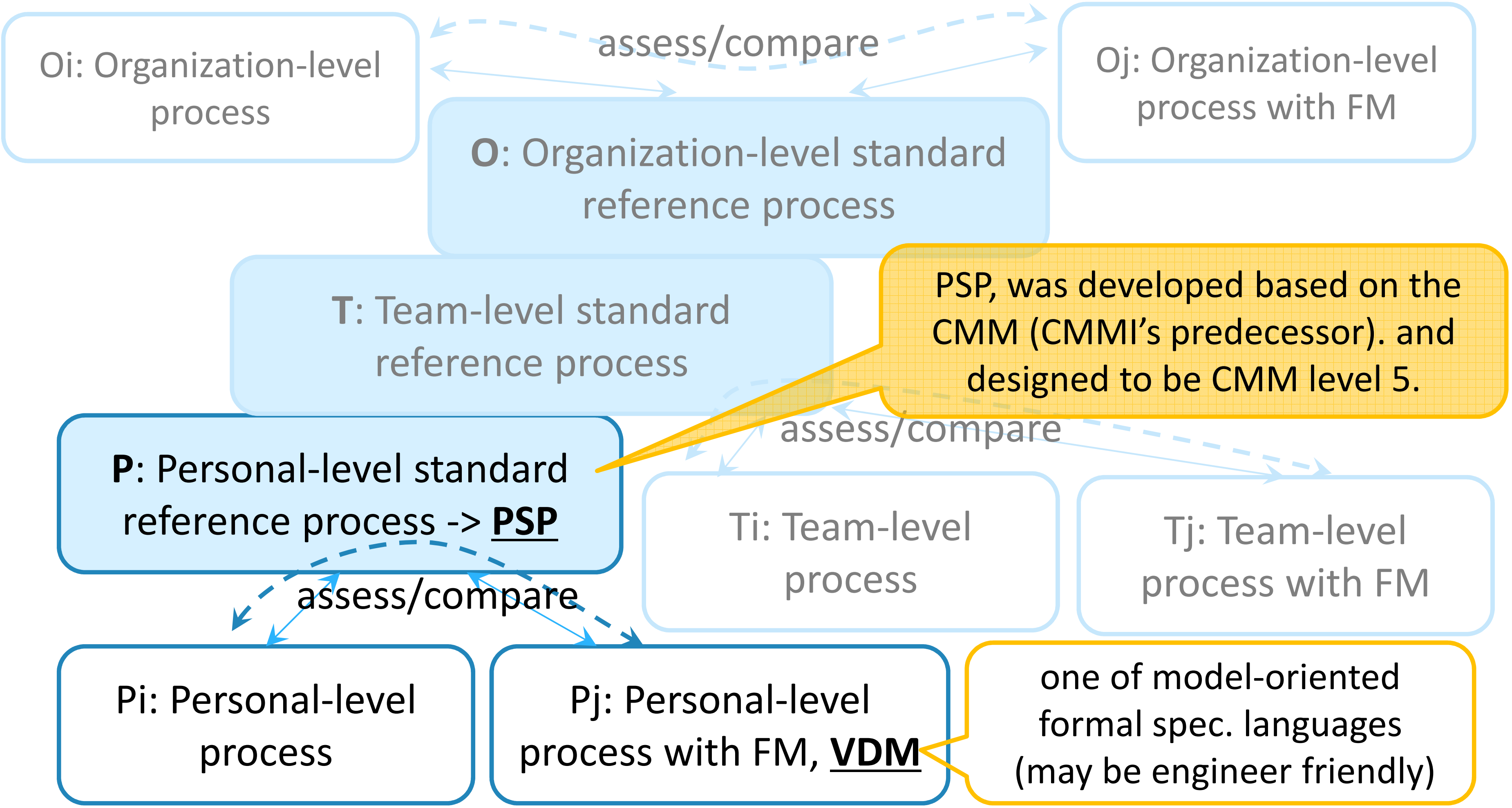
- ML5: There is a set of important Engineering PAs to continuously optimize the process.

-> analyze succeeded project instances.

Outline

- Introduction
- Process Improvement Model : CMMI-DEV
- Generating Hypotheses through the Model
- Examining Hypotheses: PSP with VDM
- Concluding Remarks

Our Early Trial: PSP with VDM



PSP: Personal Software Process*

Providing a framework that helps us to analyze where to improve our personal process:

- Phases: **plan**, **detailed design**, **detailed design review**, **code**, **code review**, **compile**, **unit test**, and post mortem, with a set of associated **scripts, forms, and templates**.
- Data: **time** and **defects** injected and removed for each phase, **size**, size and time **estimating error**, **cost-performance index**, defects injected and removed **per hour**, personal **yield**, appraisal and failure **cost of quality**, and the **appraisal to failure ratio**.

* Service Mark of Carnegie Mellon University, Software Engineering Institute

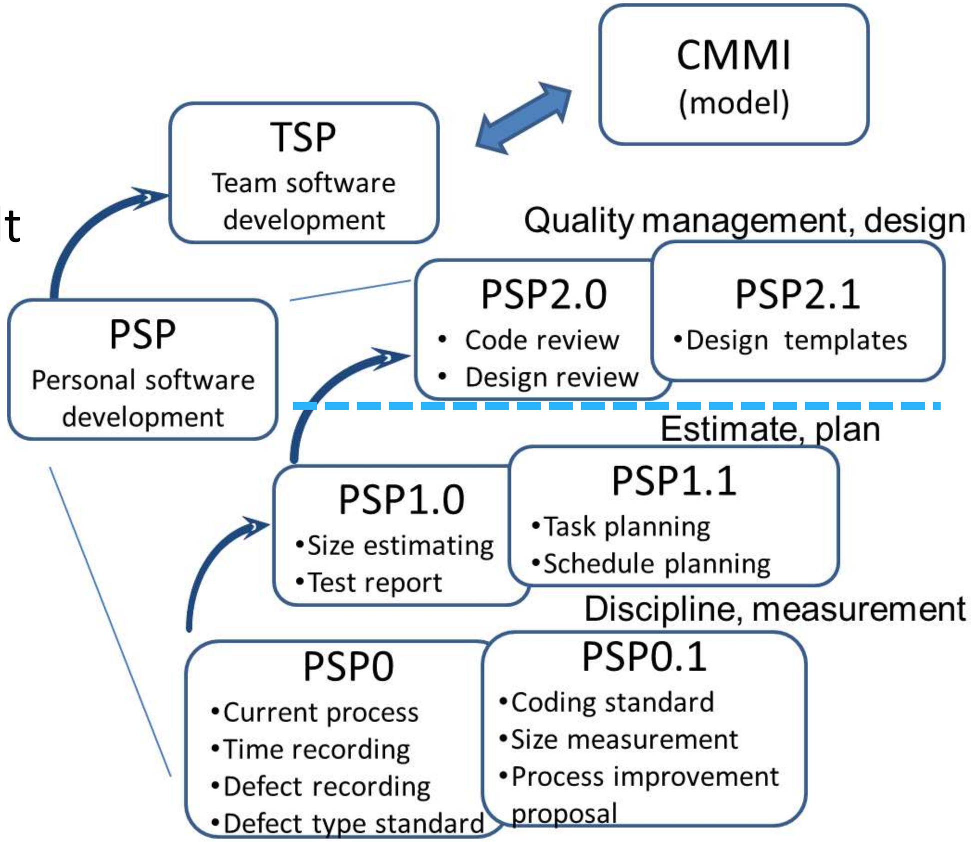
Introduce FM in PSP

PSP course structure (8-program version)

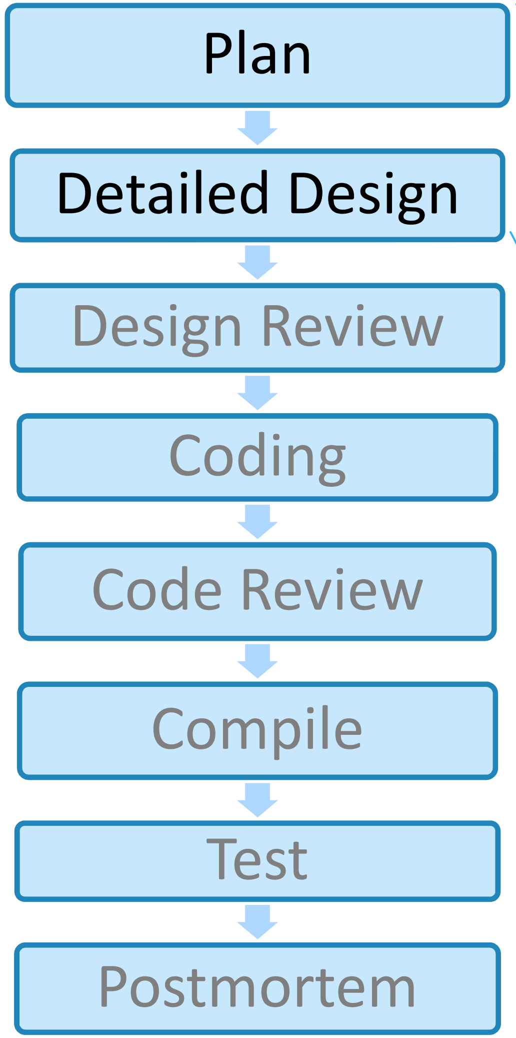
- PSP0*: measurement (2 exercises)
- PSP1*: estimate (2)
- PSP2*: quality (4)
  - simple formal notation by default

Process extension (variation)

1. Collect process data until PSP 1.1 as baseline data
   - Time, defect (type, fix time, ..)
2. Analyze baseline data and consider how to improve
3. Start using FM after PSP 2



Example: Tailoring PSP with FM



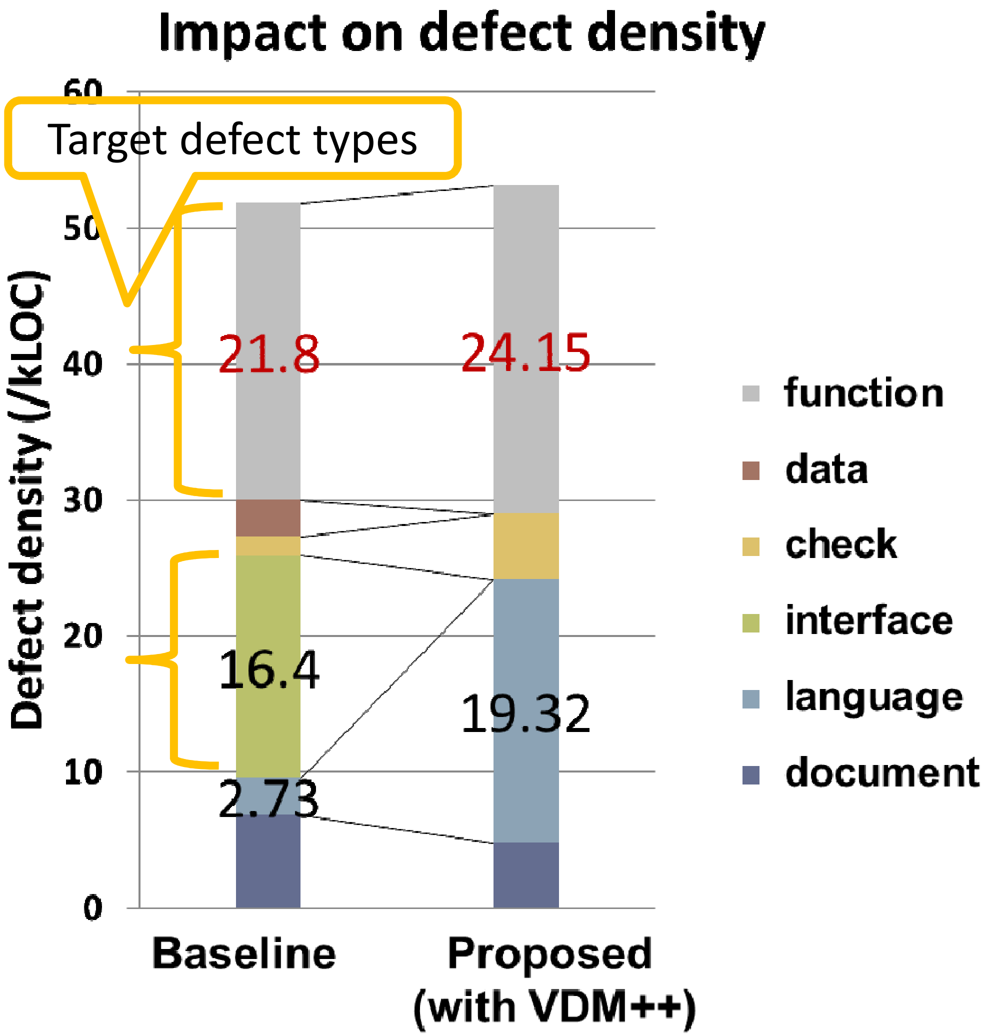
Conceptual Design

PSP Design templates

- Operational Specification template
- Functional Specification template
- State Specification template
- Logic Specification template

What if these are written in formal spec. languages

Preliminary Results: Defect Density

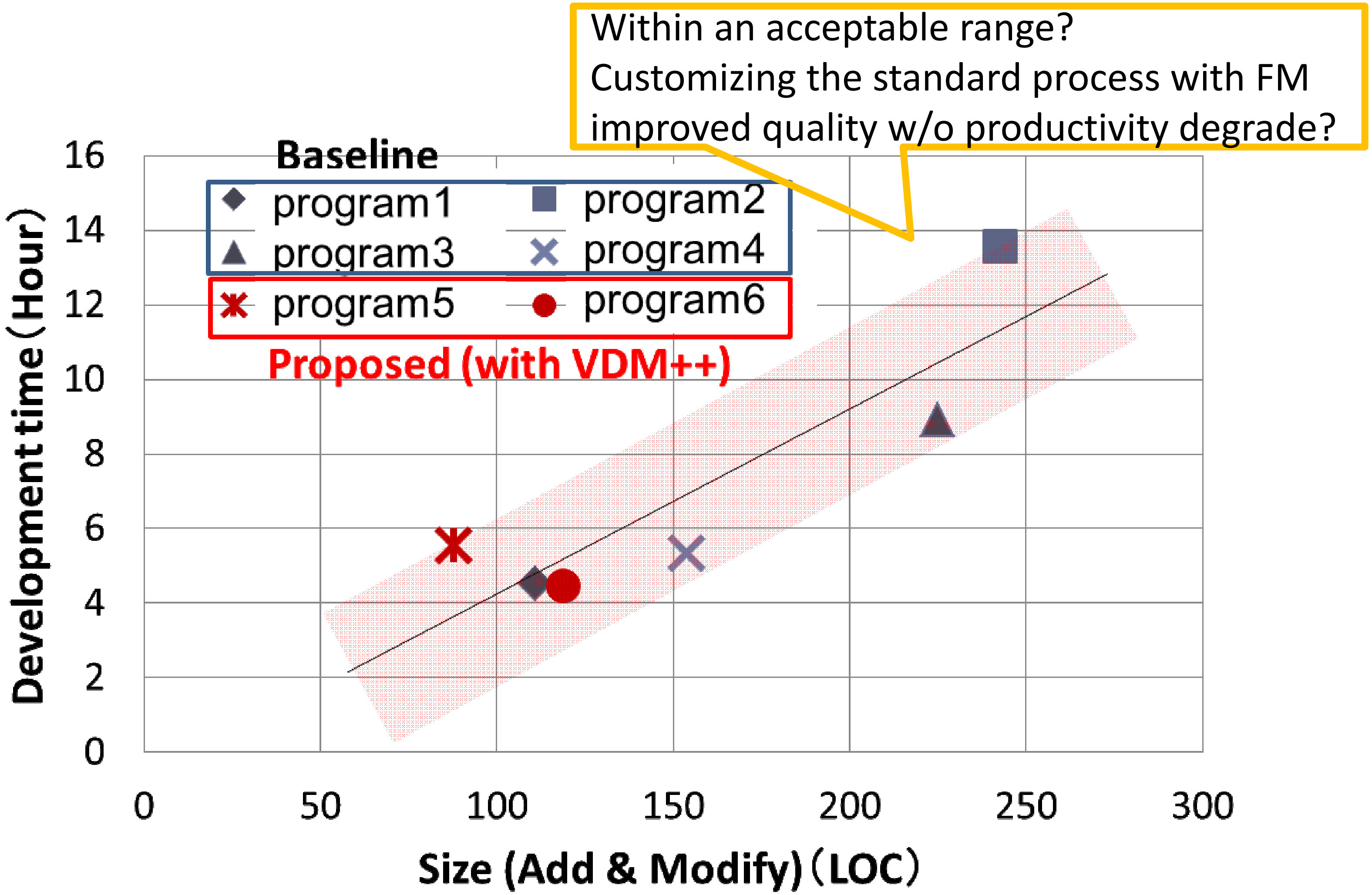


Interface type
- eliminated

Function type
(not eliminated, but ..)
- baseline
  - mainly (87.5%) removed in Test
- proposed
  - mostly removed earlier
  - only 20% in Test

Total
- no reduction ...
  - language proficiency?

Preliminary Results: Productivity



Lessons Learned

- Defined process guided introduction of FM
  - He felt that, without a disciplined process like PSP, he could not have made a process improvement plan with formal methods.
- Effectiveness in process improvement
  - He could use analyzable baseline data in improving the process
    - spent more in design and less in test
    - reduced the defects of target types without decreasing his productivity
- Current Status and Future work:
  - This was a small-sized personal-level trial.
  - > extend to scalable, generalized and proactive trial.

Outline

- Introduction
- Process Improvement Model : CMMI-DEV
- Generating Hypotheses through the Model
- Examining Hypotheses: PSP with VDM
- Concluding Remarks

Concluding Remarks

- We used a process improvement model, CMMI-DEV, as a reference in order to facilitate common understandings for the expected effectiveness in introducing formal methods into actual software development processes.
- We analyzed direct/indirect dependencies between process areas through the network analysis and generated some hypotheses.
  - We have tested / are testing the hypotheses.
- We continue more detailed analysis & evaluation with measurable and analyzable process framework such as PSP/TSP/CMMI.
- We will also apply similar approaches to other process models such as Automotive SPICE, SPICE for Space, MEDI-SPICE, SPEAK-IPA, ...