Improve through individual - TPINext and Personal Testing Process

2012-06-04 at Nordic Testing Days in Tallinn, Kari Kakkonen

knowit

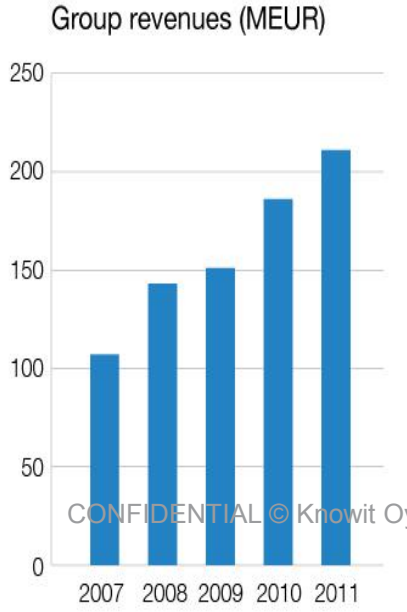
Knowit Oy - Strong resources of a Nordic Group

Knowit Oy previously known as Endero

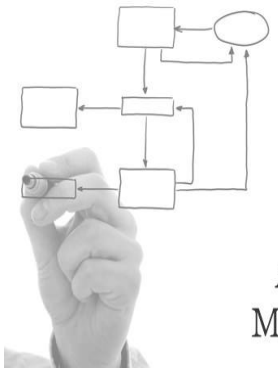
Part of Knowit Group since 2010.

1,700 experts in Sweden, Finland, Norway, Russia and Estonia.

Knowit AB is listed on the Nordic Exchange in Stockholm.



Comprehensive service offering

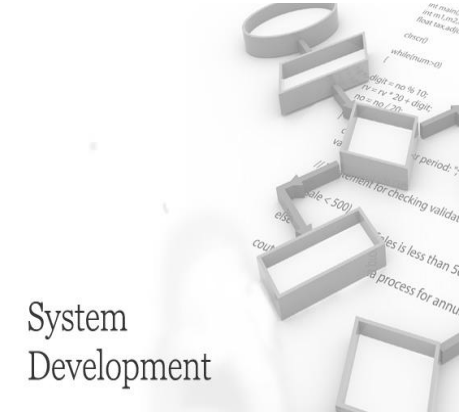


Application Management

Business & IT Management

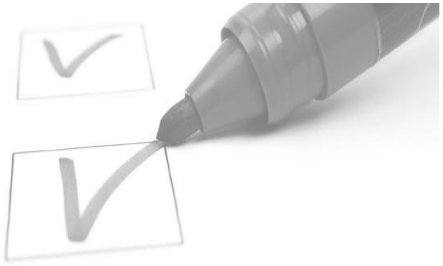
Decision

IT- & Information Security



System Development

Technology Management



Test & Quality Management

Web & Collaboration

Agenda

- TPI Next
- PTP – Personal Testing Process
- From organizational improvements to personal improvements

TPI Next

Background on TPI Next experiences

- Based on consulting work done 2002-2012 and research paper finished on Feb 2005 and its updates during 2011
- Work done at Quality Assurance Consulting Services Unit of Conformiq Software Ltd (the unit sold to Endero Ltd in June 2005) and then at Testing and Methodologies unit of Knowit (previously called Endero in Finland)
- Knowit uses currently an extended version TPI® NEXT –evaluation
- Knowit has made over 50 assessments using TPI or TPI NEXT

Needs for any test process improvements

- Objective is to identify
  - ’ok’ –areas
  - Critical improvement needs
  - Improvement needs with biggest gain
  - Smaller opportunities
- Prerequisites
  - Improvement needs to be continuous
  - Related to actual work
  - Benefit the overall process and business
- Assumption
  - Individual parts of testing process can be described and developed independently
  - Dependencies can be named and also developed independently

TPI in short

- Test Process Improvement (TPI®) is a model for improving various testing processes
- Developed and documented as a book by Tim Koomen and Martin Pol at Sogeti, the Netherlands
- Systematic way to
  - Assess software testing processes
  - Find improvement opportunities
  - Develop software testing step by step
- 20 key areas get a score based on fulfilling a number of checkpoints
  - → Improvements relate to the score

TPI Next in short

- Business Driven Test Process Improvement (TPI® NEXT) guides which Key Areas of software testing to develop
- TPI NEXT is a trademark of Sogeti Nederland B.V.
- TPI NEXT is a method to assess the maturity of an organization's or project's test process
- TPI NEXT is also a concrete and practical tool for assessing and setting goals to improve testing activities

TPI maturity example

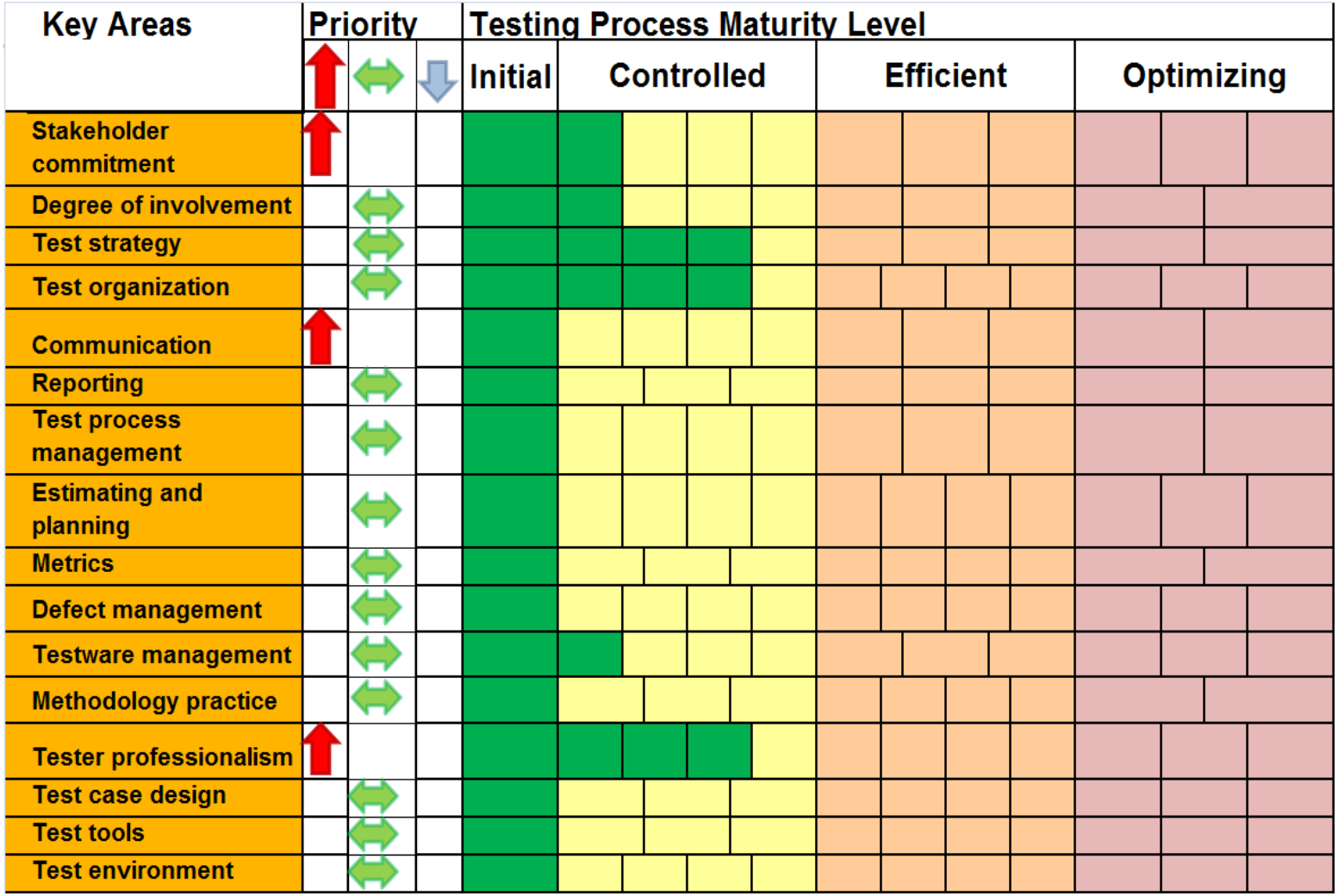
A project scores 0 to D on all 20 key areas

Key areas with leftmost A should be developed first

Key Areas		Controlled					Efficient					Optimized		
	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Test strategy		A					B				C		D	
Life-cycle model		A			B									
Moment of involvement			A				B				C		D	
Estimating and planning				A							B			
Test specification techniques		A		B										
Static test techniques					A		B							
Metrics						A			B			C		D
Test automation					A			B			C			
Test environment				A				B						C
Office environment				A										
Commitment and motivation		A				B						C		
Testing functions and training				A			B				C			
Scope of methodology					A						B			C
Communication			A		B							C		
Reporting		A			B		C					D		
Defect management		A				B		C						
Testware management			A			B				C				D
Test process management		A		B								C		
Evaluation							A			B				
Low-level testing					A		B		C					

TPI Next maturity example

- Present state of testing process



Another TPI Next example

- Scores of the key areas are not necessarily dependent on previous levels

#	Key areas	Initial	Controlled				Efficient				Optimizing		
1	Stakeholder commitment		A	B	B	C	F	H	H		K	M	M
2	Degree of involvement		A	B	C	E	H	H	J		L		L
3	Test strategy		A	A	B	E	F	F	H		K		L
4	Test organization		A	D	D	E	I	I	J	J	K	L	L
5	Communication		B	C	NA	D	F	F	J		M		M
6	Reporting		A	C		C	F	G	G		K		K
7	Test process management		A	A	B	B	G	H	J		K		M
8	Estimating and planning		B	B	C	C	G	H	I	I	K	L	L
9	Metrics		C	C		D	G	H	H	I	K		K
10	Defect management		A	A	B	D	F	F	H	J	K	L	L
11	Testware management		B	B	D	E	I	I		J	L	L	L
12	Methodology practice		C	D		E	F	H	J	J	M		M
13	Tester professionalism		D	D	E	E	G	G	I	I	K	K	M
14	Test case design		A	A		E	F	I	I	J	K	K	M
15	Test tools		E	E		E	F	G	G	I	L	M	M
16	Test environment		C	D	D	E	G	H	J	J	L	M	M

Assumptions in TPI and TPI Next

- Models seem to make some underlying assumptions, which one can see by
  - using TPI or TPI NEXT with an objective state of mind
  - scrutinizing the checkpoints, dependencies and the book
- These assumptions manifest themselves in
  - order and content of checkpoints
  - dependencies of key areas
  - improvement suggestions

Assumptions

- The assumptions seem to be that
  - A certain order of improvements is always appropriate
  - A software development organization always
    - Is from medium to large size
    - Has a separate, independent testing group
    - Is very plan-driven
    - Has fixed process phasing
    - Has testing personnel with great level of autonomy

Comparability issues

- We find that the model can be applied and adjusted in many situations
- One must take care of the comparability to other TPI or TPI NEXT -assessments
  - The flexible points in the models are when
    - It’s only more or less information
    - It doesn’t really matter
    - Things that are needed to overcome the assumptions, but don’t alter the score, in the ”spirit of TPI”
  - The corner stones that must be honored are especially
    - Development order
    - Big underlying assumptions such as independent test group
- Comparability is needed to
  - Utilize models fully – use it as intended as much as possible
  - Be able to benchmark with other TPI or TPI NEXT -assessments

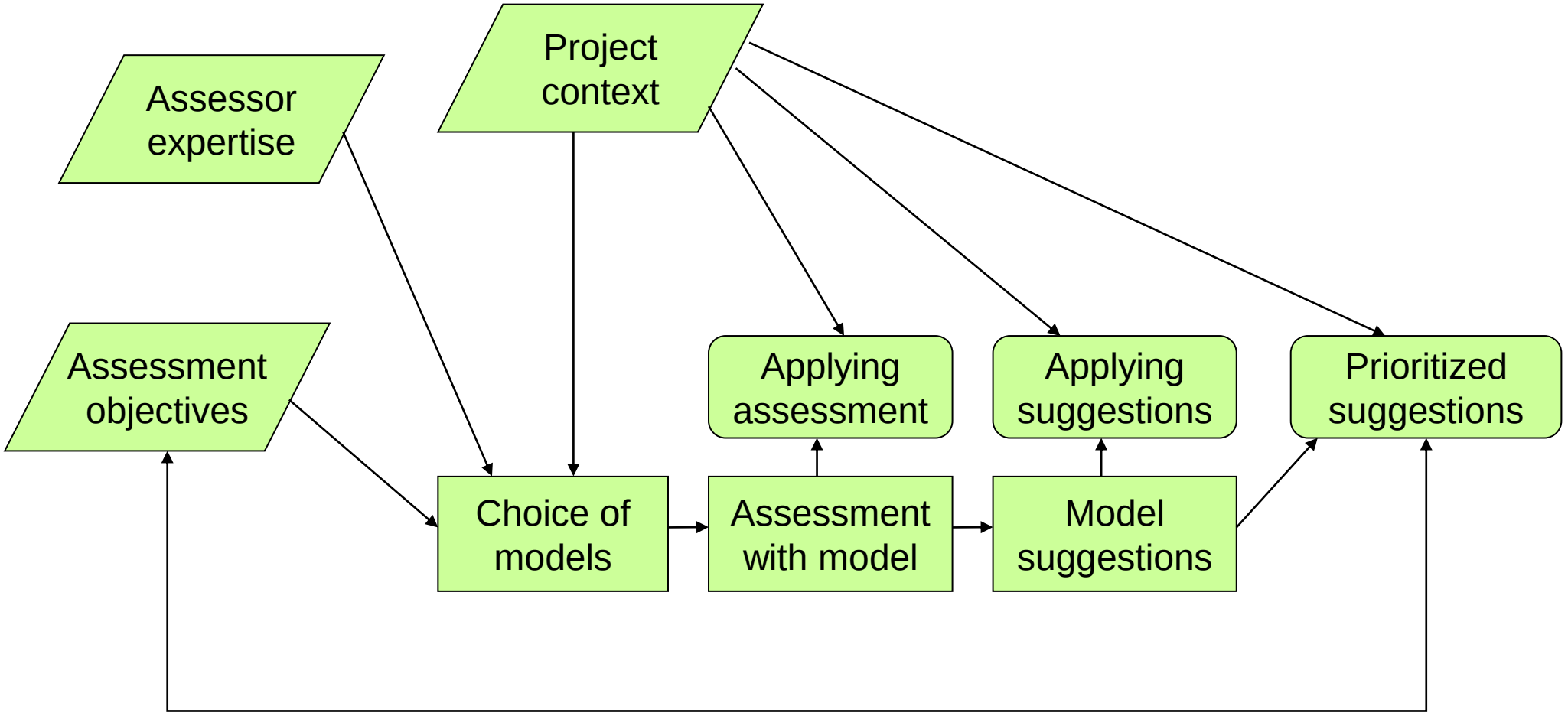
Assessment ground rules

- Scoring in key areas is only a secondary objective
- Finding improvement needs is essential
- Improvement goal has to be kept fresh in mind
- Information is needed regardless of the way one gets it – ask any questions necessary

Suggestion about possibilities of TPI and TPI Next

- Take elements or ideas from other assessment models to get fuller picture of the test situation
  - E.g. if the object is to reduce the risk, one might also ask questions inspired by Test Improvement Model (TIM)
- Apply the context-driven analysis
  - adjust thinking from single checkpoint to thinking in terms of key area level
  - based on the size of the assessed project and the process choices
- Complete the big picture of the assessor by additional questions
  - Assessor experience
  - Other models, checklists, literature
- Recognize both model-based and other realistic improvement suggestions
- Prioritize the improvement suggestions and the key areas to be improved
  - By TPI or TPI NEXT
  - By additional perspectives

Applying models



Additional questions to be used with TPI Next

- Tell us about the testing
- What kind of development life-cycle do you use? E.g. Plan-driven, Iterative, Agile? Tell us about them?
- How is the software put together? Requirements? Coding? Building? Delivery to testing? Testing?
- What are your priorities? What needs to work?
- Tells us about daily routines? Of tester, of anyone working with testing?
- What can go wrong?
- What do you think should be prioritized as improvement?

Case– with TPI Next

- Several small agile projects in a fairly large company
  - Average size 2-3 persons development effort in 1 month
  - Separate acceptance testers part timing from their normal duties
- Web-based systems and internal applications with large user base
- Quick development cycle requires lighter organization and planning but quite demanding level of software quality
- TPI Next was appropriate to find improvements although some checkpoints would automatically fail
- Focus on acceptance testing, not unit testing
- (Data partially altered to anonymize)

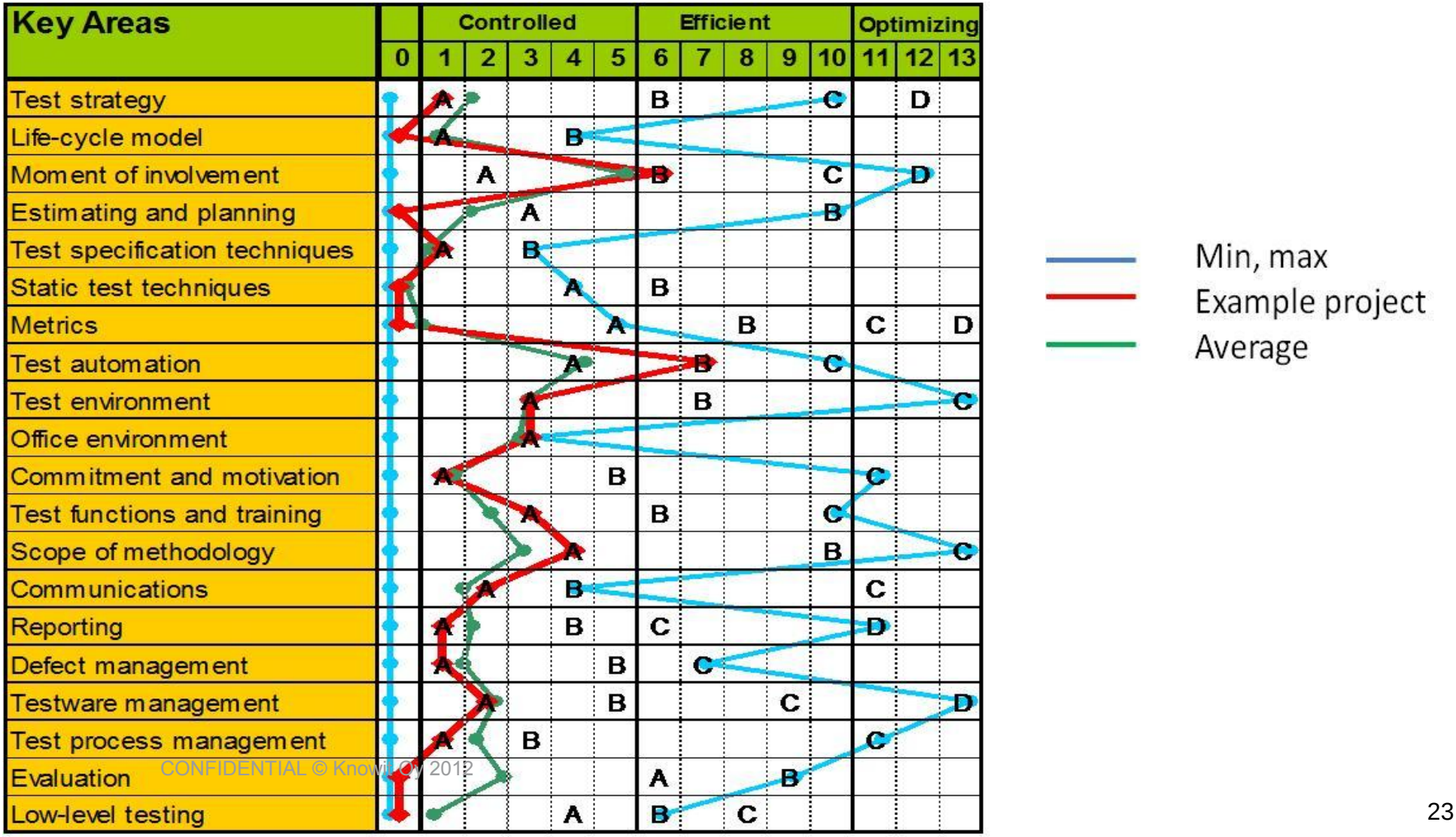
Some TPI Next adjustments made

- Full time test manager and tester requirements approved with principle ”practically full time during project”
- Any well-thought written record approved as document
  - Level of planning assessed rigorously
- Level and quality of communication assessed, not the means
- Level stakeholder involvement assessed based on second-hand information

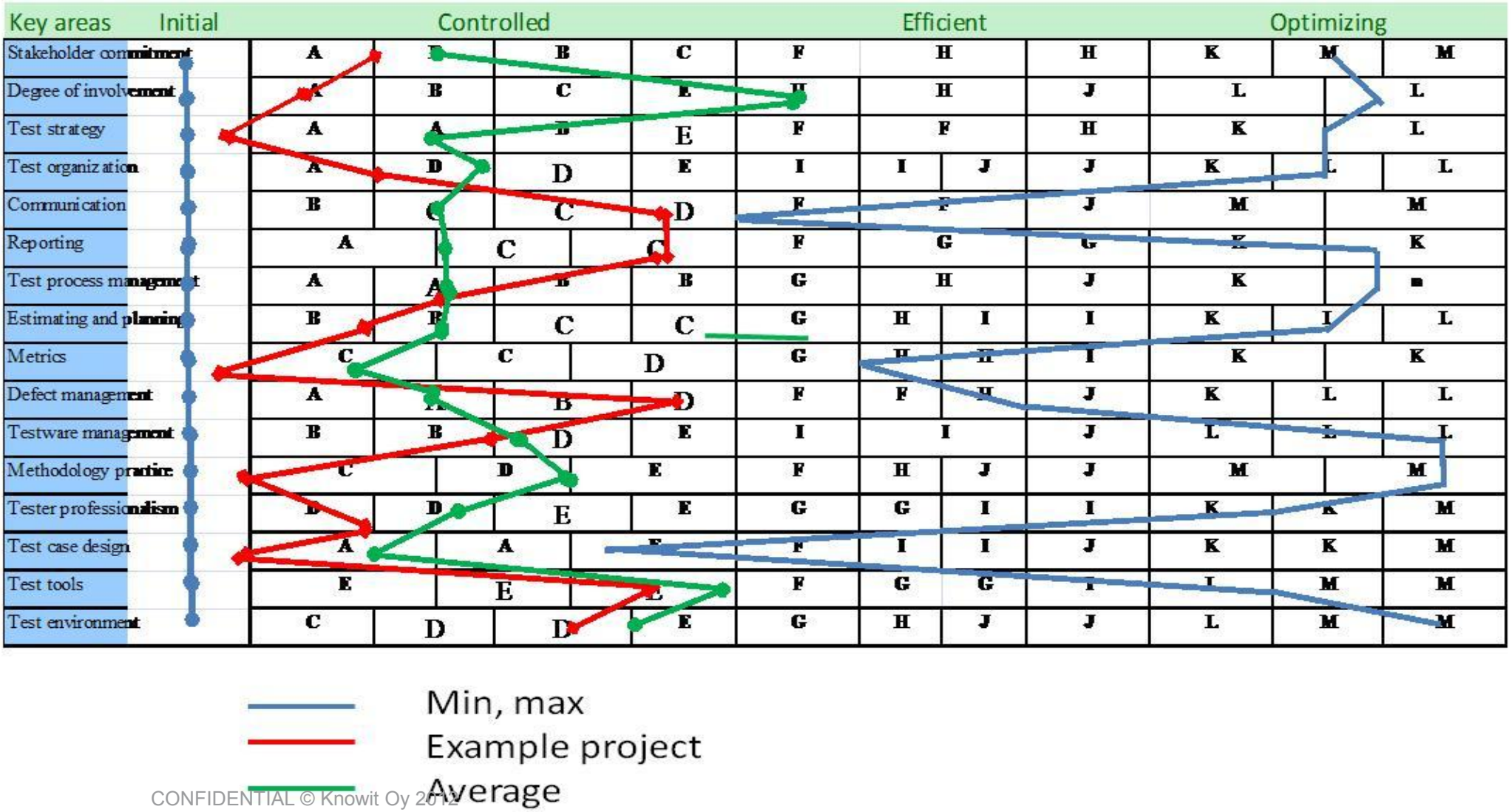
Some improvements suggestions

- Score: level Controlled to Efficient
  - Given both prioritized suggestions and ”needed for TPI” suggestions
  - Often bits of Efficient realized
- Testing environment needs a responsible support person
- Test automation usage could increase from the level of using test management tools, as there is a centralized expert team in the company
- Email communicated decisions should be stored to a central location, preferably the test management tool
- Keep the present moderate level of written planning, increase test coverage and distribute knowledge about the coverage
- Test strategy thinking should be encouraged, not to be done only as a part of process

TPI Benchmark

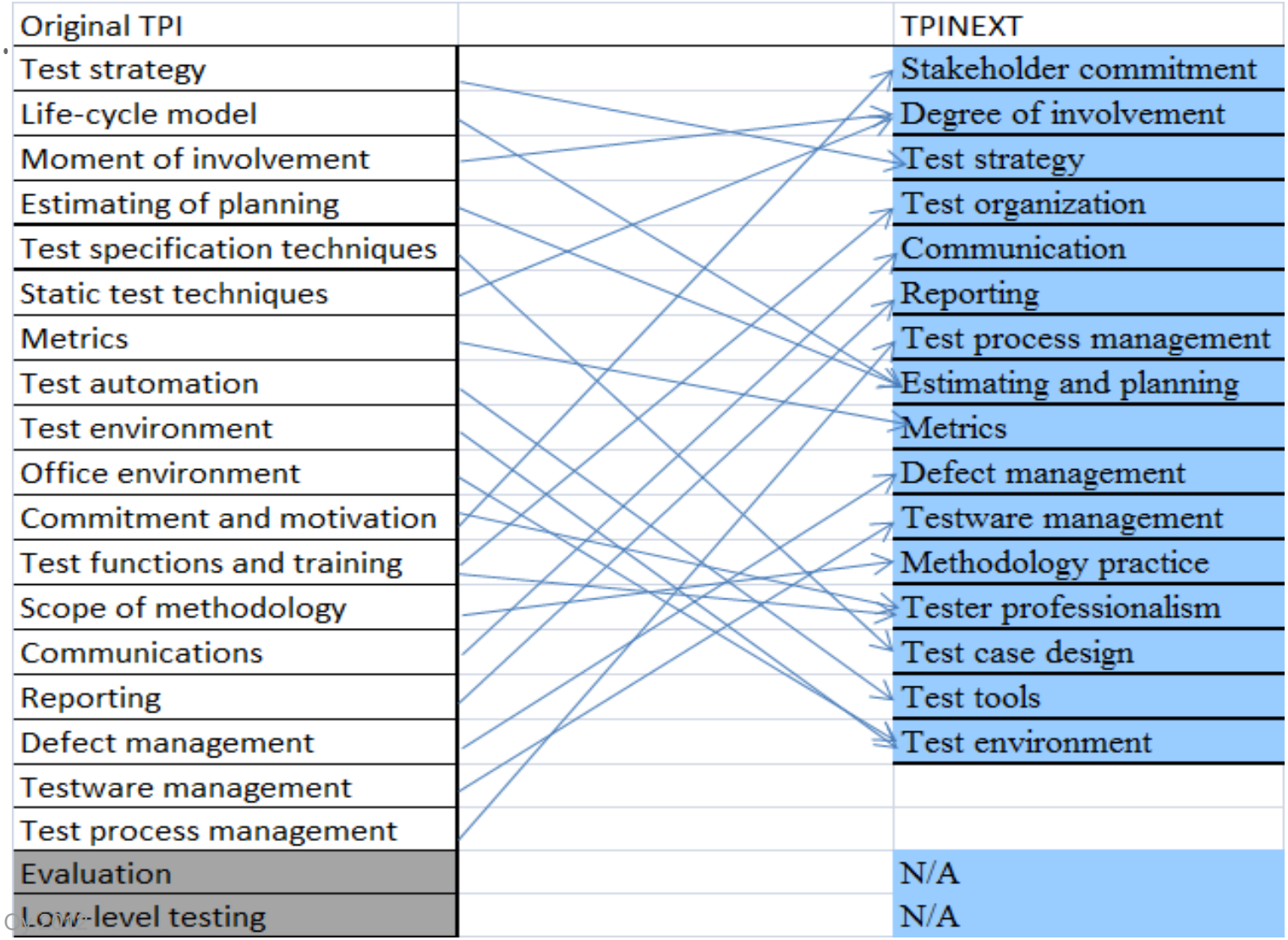


Knowit extended TPI Next benchmark



Converting TPI results to TPI Next

- Some key areas are converted directly
- Some key areas need to be weighted when TPI NEXT key area is transformed from two original TPI key areas
- Weighting can be e.g. 50/50 or 10/90



Benefits in benchmarking

- Comparing results company internal (e.g. comparing results of several projects)
- Comparing results of a company or organization to average results
- Improving and developing testing (e.g. by key areas)

PTP –Personal Testing Process

PTP Basics

- PTP is a model to measure the maturity and capability of the tester
- Inspirations:
  - Software development process assessment models SPICE (ISO 15504) and CMMI.
  - PSP (Personal Software Process) developed by Watts Humphrey
  - Bloom’s learning taxonomy.
- The model implements similar things, but in testing and on individual level

PTP utilizes Bloom taxonomy's six knowledge levels

Bloom taxonomy defines level of knowledge.

- **Remember**. Show that you know.
- **Understand**. Show that you understand.
- **Apply**. Show that you can utilize what you have learned.
- **Analyze**. Show that you can find the relevant information.
- **Synthesis**. Show that you can create new elements by combining existing ones.
- **Evaluate**. Show that you can assess ideas, information, processes and solutions.

PTP five levels

Tester is on level:

- One: Can test. (Understanding)
- Two: Manages test execution. (Apply)
- Three: Can establish personal process. (Analysis)
- Four: Measure and analyze the process. (Synthesis)
- Five: Sets goals and develops process. (Evaluate)

PTP Level 1: Can test

- Test strategy.
- Knows test types
- Knows test levels and V-model
- Know test methods and techniques
- Knows how to create a test case
- Can execute a test and log an incident
- Knows software development lifecycles.

PTP Level 2: Manages test execution

- Knows test tools and their benefits
- Knows software development design techniques e.g. UML
- Knows how to estimate test effort and create test schedule
- Tool support, measuring execution, quality assurance, test coverage
- Review of test execution
- Measurement of test execution

PTP Level 3: Personal process

- Process thinking is established
- Testing consists of chain of activities by tester: design, implement, execute tests
- Designing tests the tester
  - Groups similar tasks
  - Recognizes the groups as separate tasks: these are the activities of the process
  - Develops the activities to be regular
  - Creates constructs for the test cases: constructive testing
  - A method of working is created: Personals Testing Process – PTP

PTP Level 3: Simple example

- User interface screen with two fields and two ways to test them
- Alternative 1:
  - First the tester executes all positive tests for both fields and then all negative tests for both fields
- Alternative 2:
  - Tester executes field by field all test of that field, first all tests of field 1 and then all tests of field 2
- Tester chooses her method of working based on efficiency. Efficiency is measured.
- When the tests are done the same way the next time, a process has been established

PTP Level 3: General testing process

- Process thinking means that tester always has ”instructions” about how to work
- General testing process model can be e.g.: requirement analysis, test design, test implementation, test execution, reporting, analysis of results, retesting of incident fixes, regression testing, closure of testing
- Think how many stages the previous decision has an impact on – almost all!
- When you look the impact from the point of view of the organization’s process with multiple projects and testers, the impact is multiplied

PTP Level 4: Measure and analyze

- Level 4 is twofold:
  - Tester measures and analyzes her own activities
  - Tester requires quality from design and implementation of the system
- Measurement:
  - Measurement and its results are useful only if the measures are not changes, i.e.. Standards or methodologies are followed
- Requirements for design and implementation :
  - Clear standards and methodologies in use
  - Clear documentation

PTP level 5: Set goals and develop process

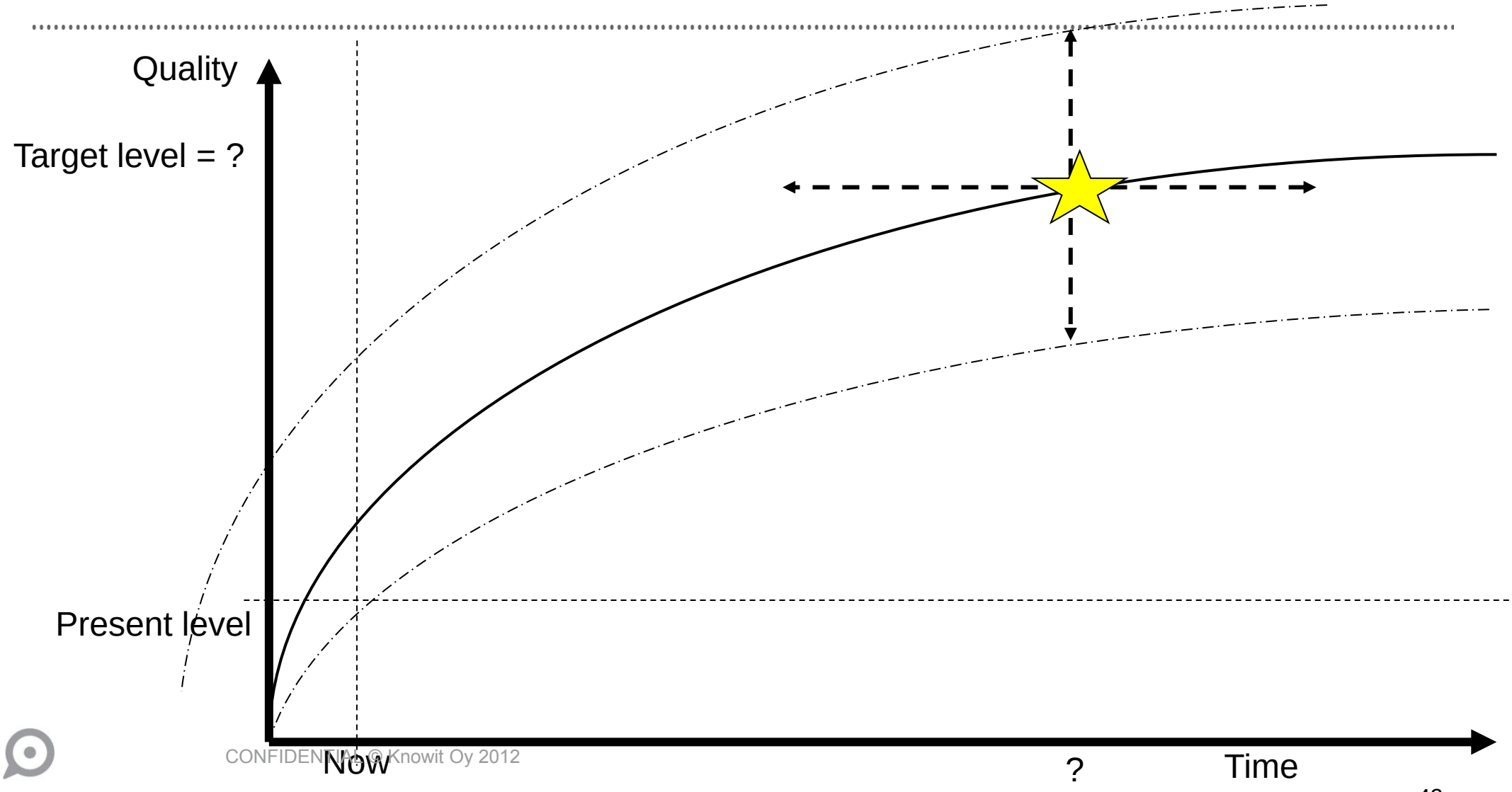
- Tester can recognize the development areas through analysis
- Sets development goals
- Finds means for development
- Can change her way of working
- Follows the effects of changes
  - Optimize and minimize process
  - Prepare for future
- Change is a continuous state
- Use effective methodologies, use information from analysis, practice, learn from others, study new methodologies

PTP Conclusion

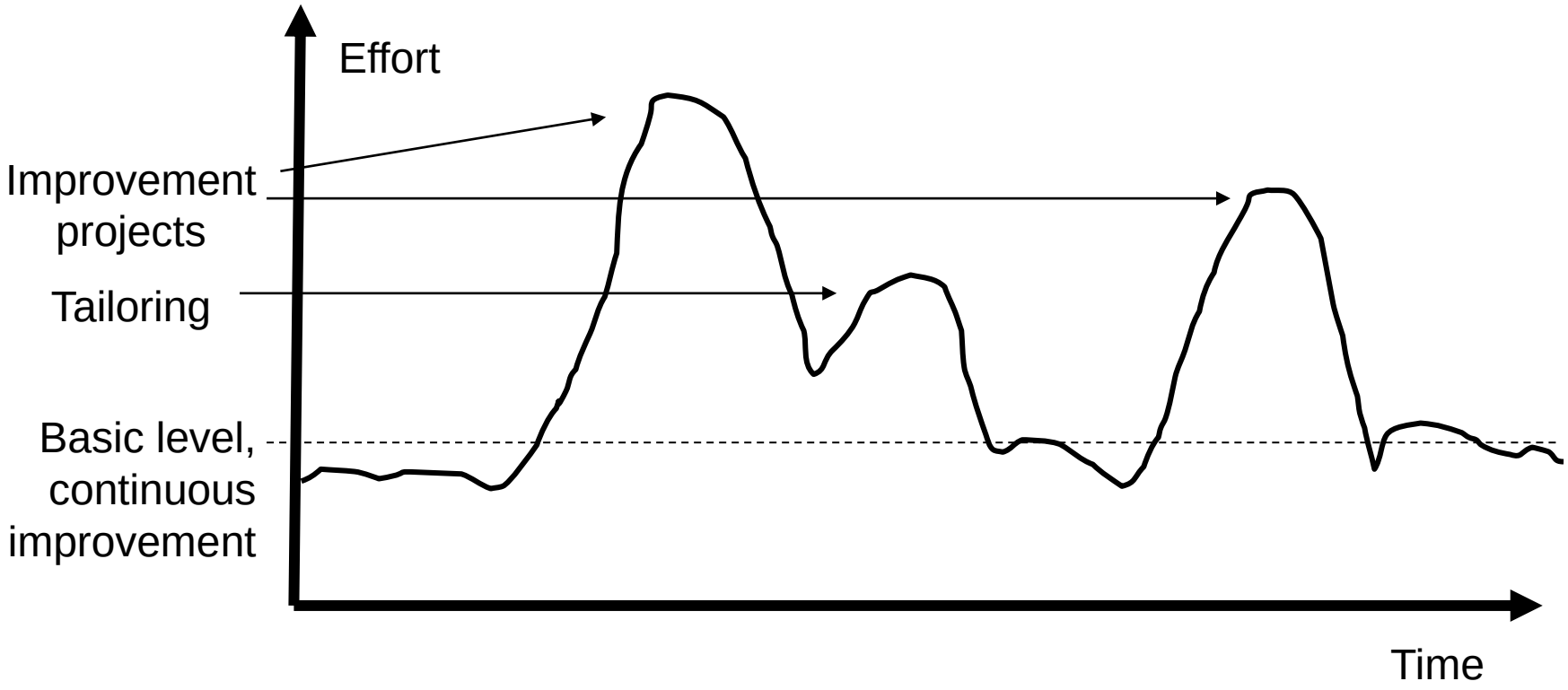
- This is a though process – you can shape your work also like this
- Two main concepts: process and staged model
- Everyone has her own ways of working even without PTP
- Tester works in different testing processes over time. The main way to become better is one’s personal process
- Recognize your own process!

From organizational improvements to personal improvements

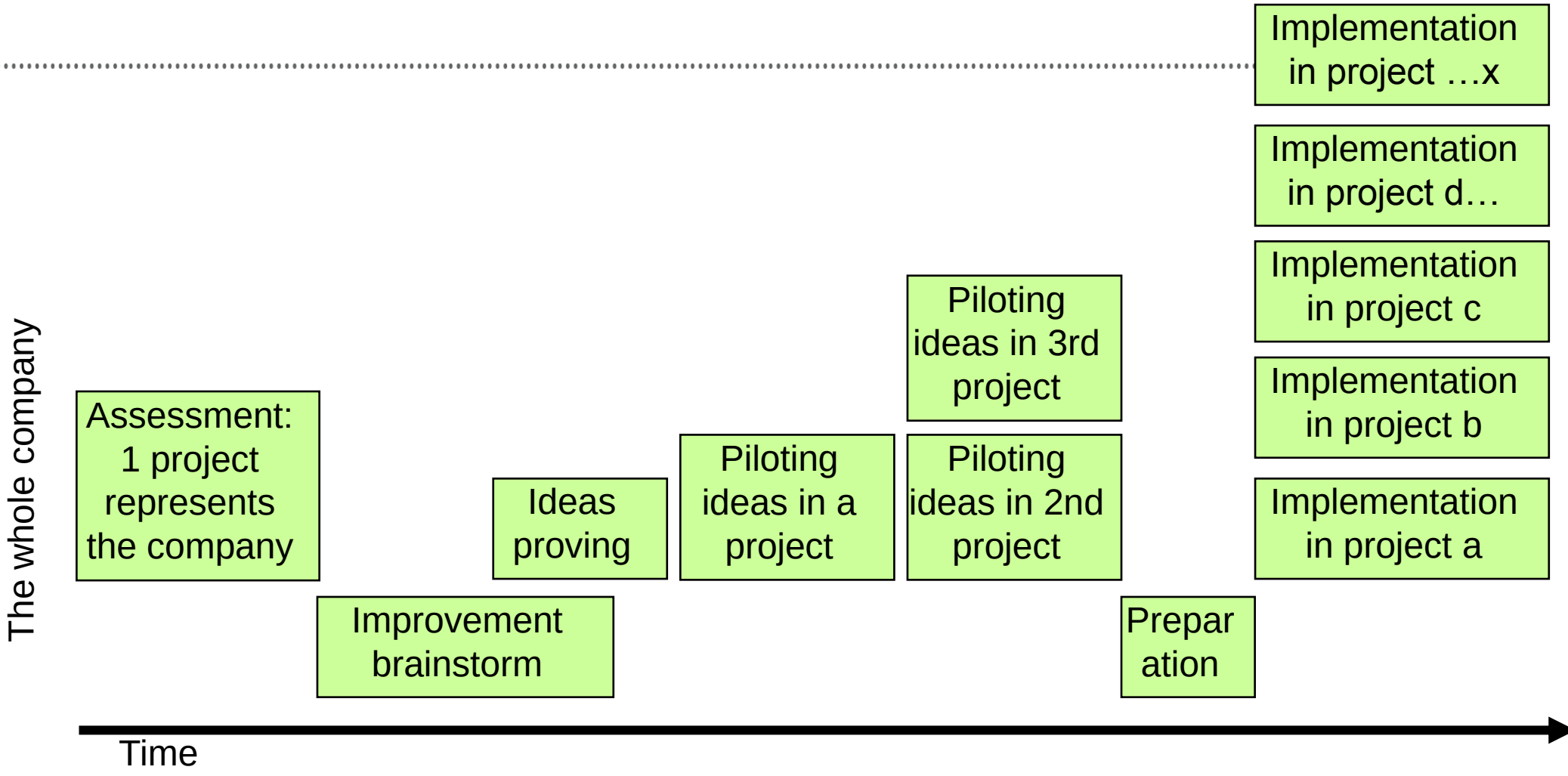
Target Level and Schedule



Level of effort



Scope of improvements



Continuous improvement

- Organizational development on different levels tends to happen in bursts
- Requirement of continuous development remains
  - Individual level is the solution
  - Empower the individual!

Levels of improvement, TPINext and PTP

	Product maturity	Process maturity	Organizational maturity
Company level	• Influenced by process	• TPI Next(by choosing representatives)	• Influenced by process
Portfolio level	• Influenced by process	• TPI Next (by choosing representatives)	• Influenced by process
Project level	• Influenced by process	• TPI Next (best suited)	• Influenced by process
Individual level	• Influenced by process	• PTP	• Influenced by process

Conclusion

- Test process development tends to focus on project or higher levels
- Individual is left with goal only – need help → Personal Testing Process
- Take care of all levels of improvement
- Process change drives product and organizational change
  - If you want a good product, have a good process
  - If you want a good organization, have a good process

Kari Kakkonen

@knowit.fi