12TH INTERNATIONAL DEPENDENCY AND STRUCTURE MODELLING CONFERENCE, DSM'10
22 – 23 JULY 2010, CAMBRIDGE, UK

PEM – A NEW MATRIX METHOD FOR SUPPORTING THE LOGIC PLANNING OF SOFTWARE DEVELOPMENT PROJECTS

Zsolt Tibor Kosztyán and Judit Kiss
University of Pannonia

Keywords: project scheduling, Dependency Structure Matrix (DSM), Stochastic Network Planning Method (SNPM), Project Expert Matrix (PEM)

1 INTRODUCTION

Nowadays the notion of 'project' is often used, however, it is important to make difference between projects. In the followings we make distinction between two groups. In the first group there are those projects (e.g construction projects) that follow the same technological order and building up from the same tasks according to a fixed order. In the second group there are projects requiring a more complexed planning (e.g. product development or software development projects), as the order of tasks can be handled in a more flexible way therefore the technological order is less fixed. In some cases the realisation of some functions/tasks can become uncertain due to time and/or resource constraint(s).

2 PROJECT SCHEDULING METHODS

Network planning methods are mainly used for planning and scheduling projects (PMBOK, 2006). However, Critical Path Method (Kelley and Walker, 1959) and Metra Potential Method (or Precedence Diagramming Method) (Fondahl, 1961) can only handle tasks with given duration, the Program (or Project) Evaluation and Review Technique can handle stochastic durations (Fulkerson, 1962), while Graphical Evaluation and Review Technique can also handle decision events (Pritsker, 1966). These methods were developed for the scheduling of traditional projects, so at product or software development projects they can only be used partially or not at all, because these network planning methods cannot handle the specialities of these projects.

2.1 The principles of matrix methods

The beginning of the development of network planning methods dates back to the 1950s, while matrix methods came to exist in the 1980s due to the work of Donald V. Steward. His Design/Dependency Structure Matrix (DSM) can be applied in different areas from system modelling to project scheduling; the practical application possibilities were primarily proven to be effective in case of product development projects (Steward, 1981). The rows and columns of the matrix show project tasks in the same order, and the marks in the matrix cells (with black or 'X') refer to the precedence relations between tasks. Initially only the evidence of strict relations was marked with the help of DSM, but these did not provide any special information about the relation (dsmweb.org).

2.2 Improved matrix methods

In case of the Numerical DSM (NDSM) the precedence relations are not just signed but also weighted according to different viewpoints. These relation/dependency weights could fall to different categories or in some cases (e.g. dependency strength) the weights are represented with numbers between 0 and 1. The bigger the value, the stronger the relation is. Primarily this method was developed for supporting system analyses or product development processes, later it was used for analysing and planning of projects as well (dsmweb.org).

During a research at the University of Pannonia a new method was developed, namely Stochastic Network Planning Method (SNPM), which is independent from the NDSM. It is similar in appearance but different in semantics. In the SNPM the importance of a relation is represented with a value between 0 and 1. 0 shows a total independence, 1 shows certain relation, and the value between 0 and

1 shows uncertain or possible relation between tasks. If the values of cells are probabilities of the relation instead of the importance, it is signed with $p_{(\mathbf{A},\mathbf{B})}\in[0,1]$ in case of tasks **A** and **B**, then $1\text{-}p_{(\mathbf{A},\mathbf{B})}$ shows the probability that there is no relation between the two tasks. If $1\text{-}p_{(\mathbf{A},\mathbf{B})}=p_{(\mathbf{A},\mathbf{B})}=0{,}5$, then the relation between these two tasks is indifferent, so the tasks can be realised sequentially and parallelly with the same probability (Kosztyán et al., 2008).

Relations between tasks can be treated as probability if there are some a priori information about possible technological order from similar projects which were realised previously (in this case they are objective probabilities); or rather possible technological relations can be formed based on the opinions of different experts (in this case they are subjective probabilities). The diagonal does not play a role, so the values in the diagonal cells can be signed with 0 or with empty cells as well (Kosztyán et al., 2008).

Since the relations are weighted according to their importance, it can be assumed that different graphs can be depicted from the matrix as a result. According to the importance of the relations it can be decided whether the realisation or the omission of the possible relation is more practical. Prior experience of similar projects, the constraints, the objective function (e.g. the most occurrence project scenario, minimal lead time, the least resource using, or combination of multiple objective functions) can influence this decision.

2.3 Specialities and application possibilities of PEM

SNPM can be used in many cases, however, this method cannot handle all problems. For example in IT projects, especially in the case of software development it is possible that the order of some tasks can be reversed or tasks can be left out or replaced with other tasks (Kiss and Kosztyán, 2009a). These cases cannot be represented in SNPM, because this method can "only" handle the possible relations. We enhanced the SNPM and created the Project Expert Matrix (PEM). PEM can handle the possible occurrence of tasks as well, because the importance/probability of the task realisations can be represented in the diagonal of the matrix. Mark 'X' or 1 shows the certain tasks, the value between 0 and 1 shows the uncertain or possible tasks.

If the values in the PEM diagonal show the (relative) priority/importance of the realisation and information about the cost, time and resource need are given, then the following exercise can be defined: A project scenario has to be determined that includes the most tasks within the given time, cost and resource limit. This process can be interesting in case of the so-called agile project planning that is used primarily at IT development projects (Kiss and Kosztyán, 2009b). In this paper we propose a solution to this problem.

The agile project planning technique used at IT projects puts the concept of project management up-side-down. At the traditional project planning the goal of the realisation and the tasks are given, so the challenge is to determine the project scenario with the smallest cost, resource and time need. At agile projects the constraints are the time, the cost and the resources, while the goal is to realise as much of the tasks as possible (Dalcher, 2009).

The agile project planning lacks a comprehensive support methodology as well as software support. It is difficult to use the traditional network planning methods, because they cannot handle the possible tasks. However, PEM can help the project experts to set the importance of the task realisation and to determine the omittable task.

In the analysing phase of IT software development projects the so-called MoSCoW Analysis is a frequently applied method (Tierstein 1977). With MoSCoW Analysis those tasks/functions are defined that

- have to be realised certainly, because it is the condition of the contract (Must have);
- are not parts of the conditions of the contract, but they can be realized with a later modification, or can contain useful functions (Should have);
- although can be realized, but they require either too much cost/resources or too much time (Could have).

This analysis includes not only the tasks above (marked with M, S and C), but also those tasks, that will not realise in this project (Won't have). Value 1 shows the certain tasks, value above 0.5 shows the tasks which have to be realized practically, value below 0.5 shows the omittable tasks and 0 shows the not-realising tasks in this project. It is possible to rank the tasks according to their importance.

The values of PEM can be calculated with taking logic plans of previous similar projects or experts' opinions into account. If tasks are determined based on previous projects, then the occurrence probabilities of tasks are objective; and if the occurrence of tasks are determined based on the experts' opinions, then the occurrence probabilities of tasks are subjective. At summarization of the different plans into PEM matrix it is practical to use the geometric mean instead of the arithmetic mean because of the independence of these opinions and experience.

Diagonal values of PEM can be determined using votes employed in complex group decision making methods. These votes can be derived from the opinions of experts inside or outside of the company or from the stakeholders of prior projects as well. The votes can be calculated with equal or with different weights depending on various viewpoints (e.g. constraints or objective functions). Votes can be summarized with the help of complex group decision making methods, like KIPA or AHP (Analytic Hierarchy Process).

2.4 Determining possible logic plans from the PEM

A two-step algorithm was developed to determine all possible deterministic solutions represented by the DSM matrices or in graphs which stem from the PEM that includes stochastic tasks and relations. The uncertainty of the PEM derives from the possibility of some tasks and relations, because a possible task and relation between tasks can be either realised, or not. If it is realised, then the occurrence probability of the project scenario is calculated with the value in the matrix (p), and if not then the occurrence probability is calculated with the complement ($1-p$).

Firstly the PEM is regarded on the level of the tasks so all possible solutions have to be determined focusing on the values in the diagonal of the matrix. All possible combinations have to be created from the PEM, where each possible task can be realised (1) or not (0). This way SNPM matrices or project scenarios can be determined.

Secondly SNPM matrices are regarded on the level of the precedence relations of which values are in the off-diagonal cells of the matrix. All possible combinations have to be created to each SNPM matrix to determine the case whether there is relation between two tasks (1) or whether there is not (0). These possible combinations can be represented with DSM matrices and/or representation graphs as well. In this way DSM matrices or project structures are determined.

When determining the project scenarios it is necessary to define the tasks that have to be realised within a given time, cost and resource limit. It is the answer to the question: **WHAT** are those tasks which have to be realized in the course of the project. If the project scenario is determined, another question occurs: **HOW**, in what kind of logic order has the project be realised.

2.5 Selection methods

Some algorithms were developed for ranking the possible solutions and choosing the best solutions. The Project Scenario/Structure Selection Method (PSSM) begins with the definition of tasks, and then they can be ranked according to their importance/probabilities in descending order. Since the tasks signed with 1 are the certain ones and the tasks with priority/probability 0 will not be realised, these tasks do not influence the number of possible project scenarios. It depends on the number of the uncertain/possible tasks (S and C). If it is k, then 2^k is the number of possible project scenarios. The selecting process of the best project structure from the project scenarios is the same process as at PSSM.

PEM can be extended, so more data can be depicted simultaneously assigning to the tasks (importance/probability, duration, resource and cost needs) and to the relations (importance, possible delay). If the exercise is to determine the logic plan with the highest priority/probability within a given time and resource limit, then time and resource needs are the constraints. The primary objective function is the determining of the project structure with the maximal (relative) priority/probability. The secondary objective function is selecting the project structure with the maximal (relative) priority within the time and resource constraints. The possible project scenario/project structure is determined with the help of PSSM method. A new agile project scheduling method (APS – Agile Project Scheduling) can take multiple objective functions into account at the same time, so it can determine the (optimal) resource allocation within time, cost and resource constraints.

3 SUMMARY

In this paper a new matrix approach of project planning techniques was introduced. PEM can model the previous network planning procedures; however, the real advantage is when we apply this method for projects that have tasks with flexible/possible technological order. These are the IT, software development and innovation projects. The introduced algorithms can be very useful for planning and tracking these projects. These project scenarios can be ranked according to their occurrence priorities. PEM is a useful tool for project managers to handle the possible and omittable tasks, moreover it can determine the ways of the realisation of tasks. This method can (re)use the previous successful project scenarios as a part of a project expert system. It could give valuable information to project managers how to solve a given problem. The figure below shows a summarisation of the introduced procedures.

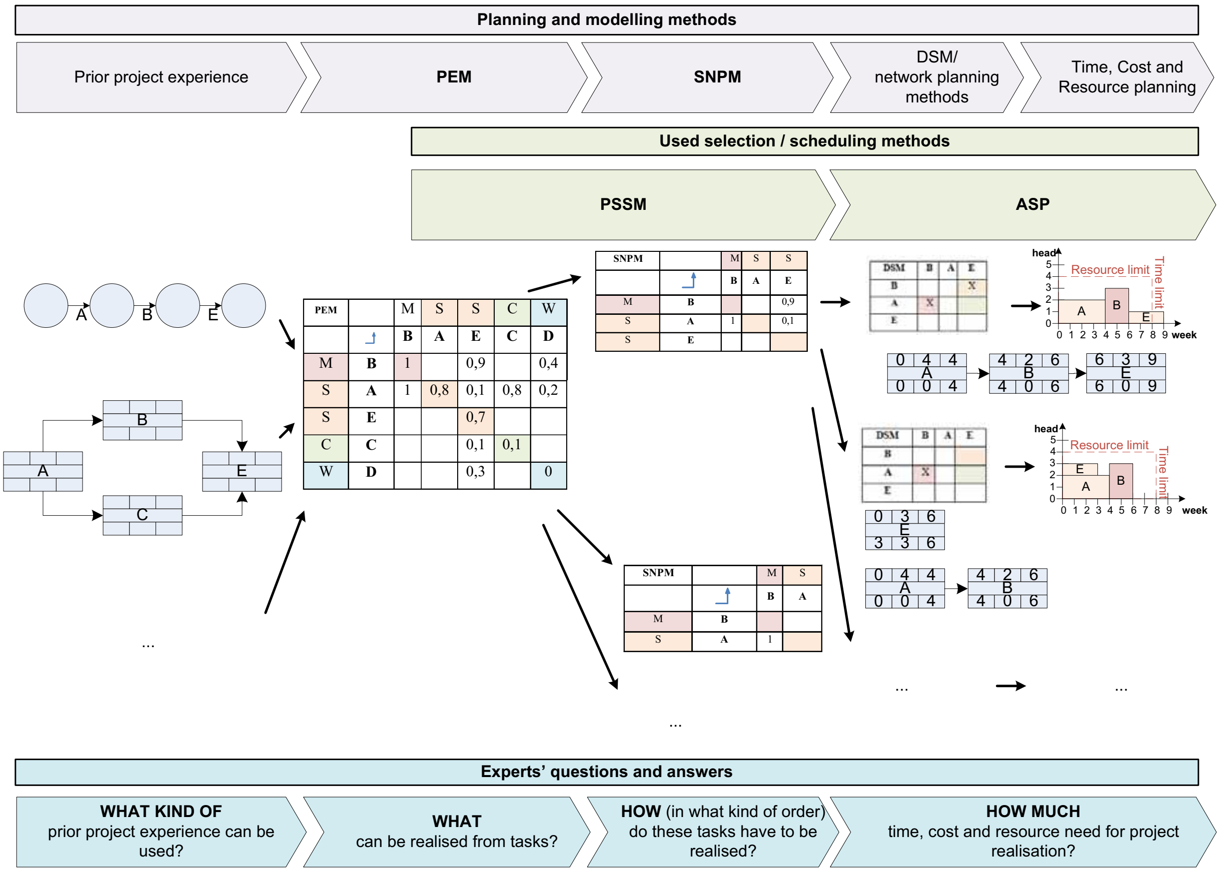


Figure 1. The summary of our methods

A program using genetic algorithms was developed at the University of Pannonia for supporting the procedures with computer. This program can handle project scenarios in case of high number tasks as well (Borbás, 2010).

ACKNOWLEDGEMENT

The authors would like to thank Edit Komlósi, Anikó Németh, Péter Cserti and István Borbás for their support.

REFERENCES

Borbás, I. (2010). Development of genetic algorithm in MOGAlib framework to support matrix based project scheduling, Master thesis, supervisor: Balázs Gaál, Department of Electrical Engineering and Information Systems, University of Pannonia, Veszprém, Hungary.

Dalcher, D.J. (2009). AiPM Book Series & Research at the NCPM, *PMUni Conference*, Vienna.

Fondahl, J.W. (1961). A non-computer approach to the critical path method for the Construction Industry, *Tech. Rep*. No. 9, The Construction Institute, Stanford University, CA.

Fulkerson, D.R. (1962). Expected critical path length in PERT network. *Operations Research*, Volume 10, Issue 6, 808–817.

Kelley Jr., J.E. and Walker, M.R. (1959). Critical Path Planning and Scheduling: An Introduction. Mauchly Associates, Ambler, PA.

Kiss, J. and Kosztyán, Zs.T. (2009a). The importance of logic planning in case of IT and innovation projects, *AVA (International Congress on the Aspects and Vision of Applied Economics and Informatics)*, Debrecen.

Kiss, J. and Kosztyán, Zs.T. (2009b). Handling the Specialties of Agile IT Projects with a New Planning Method, *CONFENIS (The Enterprise Information Systems International Conference on Research and Practical Issues of EIS)*, Győr, Hungary.

Kosztyán, Zs.T., Fejes, J. and Kiss, J. (2008). Handling stochastic network structures in project scheduling, *Szigma* XXXIX. pp. 85-103.

MIT DSM Research Group (2005). MIT DSM Web Site, http:// www.dsmweb.org/

Pritsker, A.A. (1966). GERT: Grafical Evaluation and Review Technique, MEMORANDUM, RM-4973-NASA.

Project Management Institute (2006). *A Guide to the Project Management Body of Knowledge* (PMBOK® Guide – 3rd Edition), ISBN 963 05 8401 8.

Steward, D. (1981). *System Analysis and Management: Structure, Strategy, and Design*, New York: Petrocelli Books

Tierstein, L.M. (1997). Managing a Designer/2000 Project. New York Oracle User Group. Fall '97. http://www.wrsystems.com/whitepapers/managedes2k.pdf. Retrieved 2008-05-31.

Contact: Judit Kiss
University of Pannonia
Department of Management
Egyetem utca 10.
8200 Veszprém
Hungary
Phone: +36/88/624-243
Fax: +36/88/624-524
E-mail: kissjudit@gtk.uni-pannon.hu

PEM – A New Matrix Method for Supporting the Logic Planning of Software Development Projects

Zsolt Tibor Kosztyán
Judit Kiss

University of Pannonia, Hungary

Technische Universität München
UNIVERSITY OF CAMBRIDGE

MANAGING COMPLEXITY
BY MODELLING DEPENDENCIES

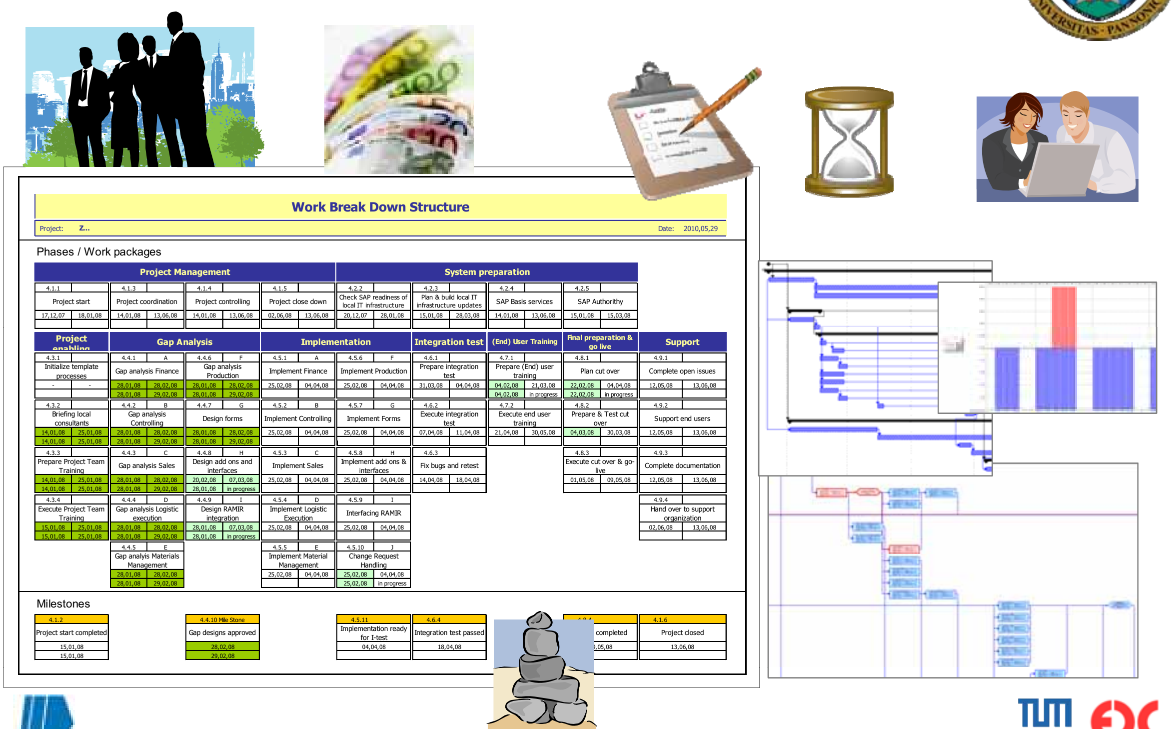
Index

Technische Universität München

Specialities of IT projects

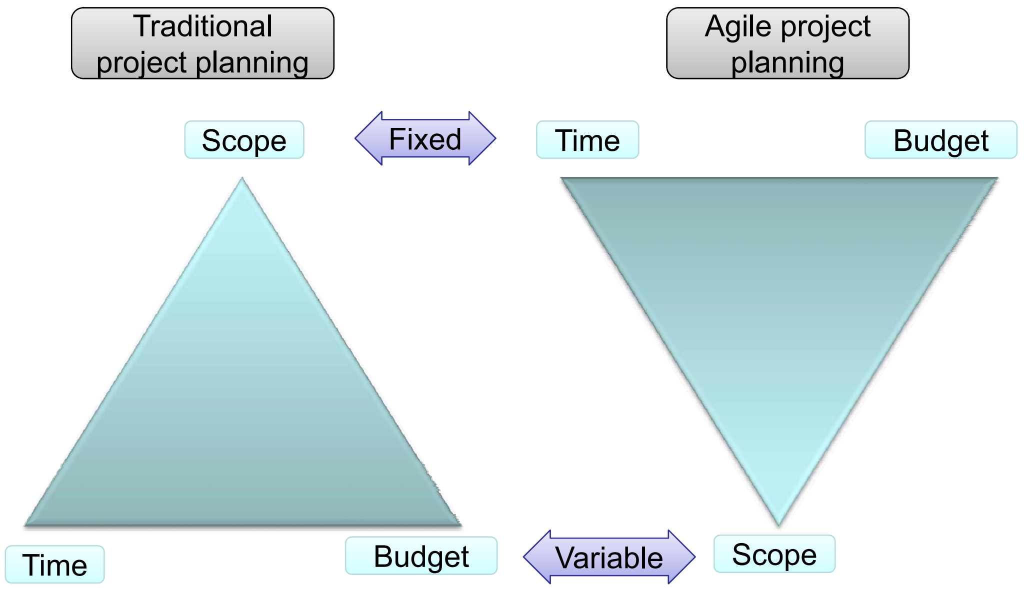
- Stochastic tasks with stochastic durations
- Stochastic relations between tasks
- More possible project structures
  - Tasks can be repeated or task sequences can be reversed
  - Flexible order of task sequences,
  - Several tasks can be realized parallelly and also sequentially
- At logic planning prior experience can be reused
- More possible project scenarios
  - Realizing tasks can be ranked by their importance
  - Less important tasks/functions can be left out from the project

Process of traditional project planning



MANAGING COMPLEXITY
BY MODELLING DEPENDENCIES

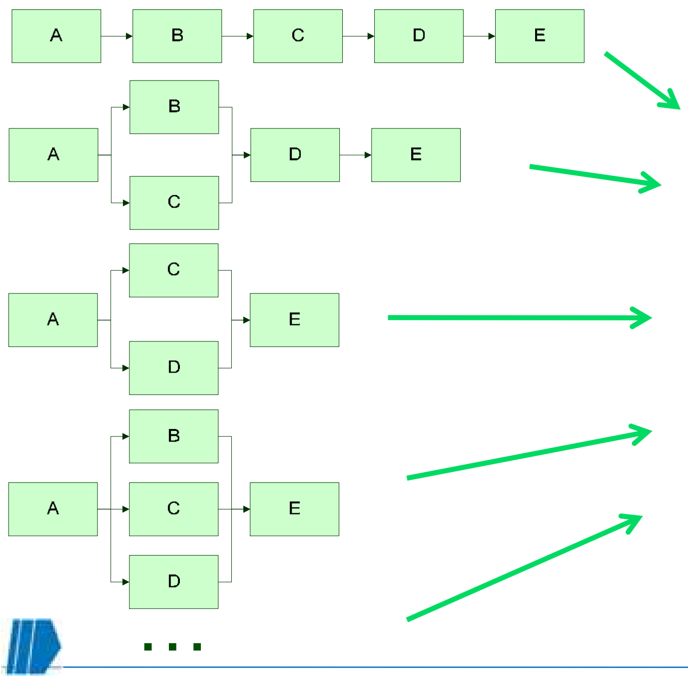
Agile project management approach for supporting IT projects



(Dalcher, 2009)

MANAGING COMPLEXITY
BY MODELLING DEPENDENCIES

Storing prior project experience



Project Expert Matrix

PEM	A	B	C	D	E
A	**1**	0.9	0.7	0.3	0
B	0	**0.8**	0.4	0.6	0.25
C	0	0	**1**	0.5	0.5
D	0	0	0	**0.3**	1
E	0	0	0	0	**1**

How can values of the PEM be determined?

- According to <u>prior experience</u>
  → *objective* probabilities
- According to <u>experts' opinions</u>
  → *subjective* probabilities
- Priorities
- Importance
- Votes
- Categories

PEM	A	B	C	D	E
A	**1**	0.9	0.7	0.3	0
B	0	**0.8**	0.4	0.6	0.2 5
C	0	0	**1**	0.5	0.5
D	0	0	0	**0.3**	1
E	0	0	0	0	**1**

Categories of the tasks

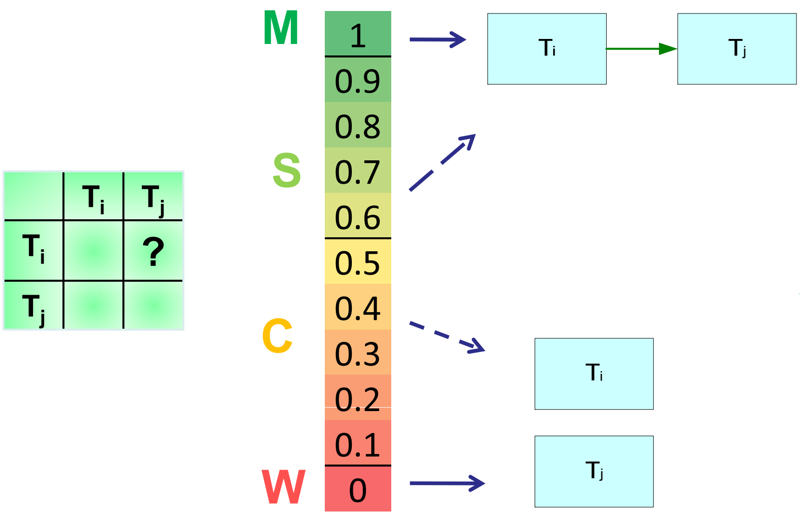
MoSCoW

analyses

	A	B	C	D	E	F
A	**1**	0.9	0.7	0.3	0	0
B	0	**0.8**	0.4	0.6	0.25	0
C	0	0	**1**	0.5	0.5	0
D	0	0	0	**0.3**	1	0
E	0	0	0	0	**1**	0,3
F	0	0	0	0	0	**0**

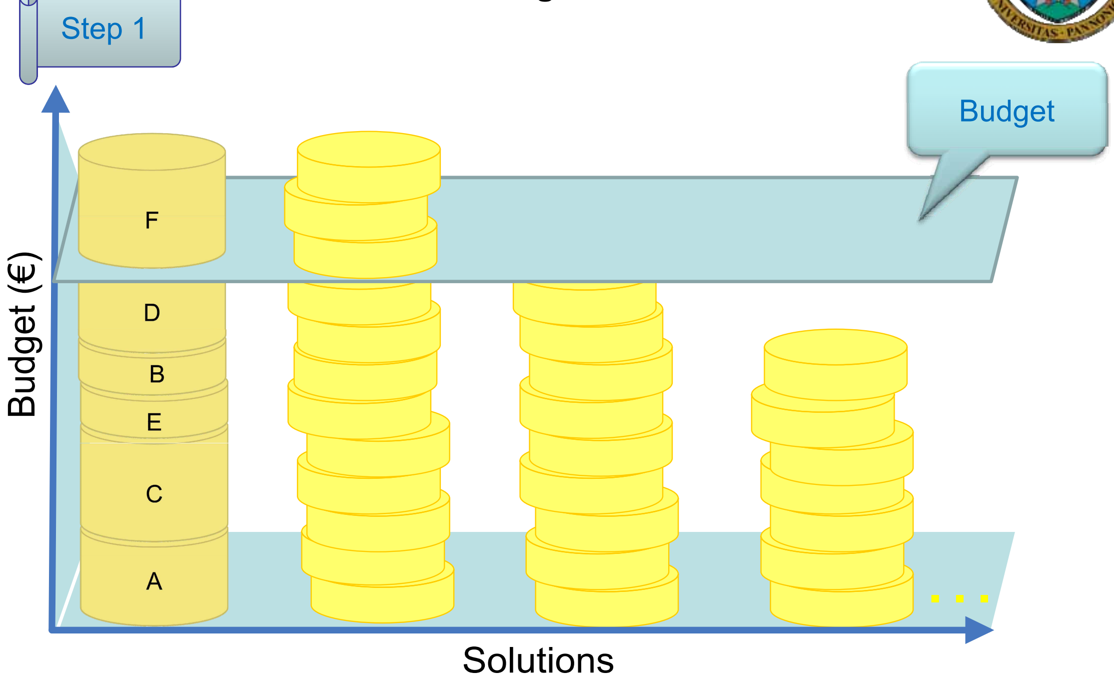
Must have	Should have				Could have					Won't have
1	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1	0

Sequentially or parallelly?



	A	B	C	D	E
A	1	0.9	0.7	0.3	0
B	0	0.8	0.4	0.6	0.25
C	0	0	1	0.5	0.5
D	0	0	0	0.3	1
E	0	0	0	0	1

Selecting the tasks



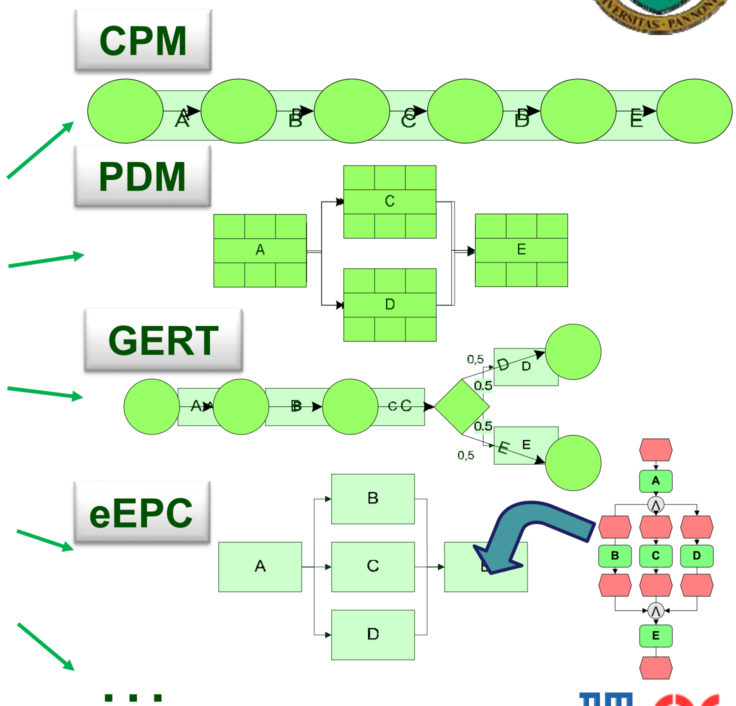
Selected tasks: A, C, E, B, D

Sharing prior project experience

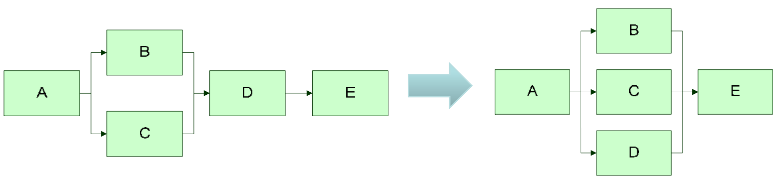
Step 2

Generating <u>all possible project structures</u> based on the *given relations*

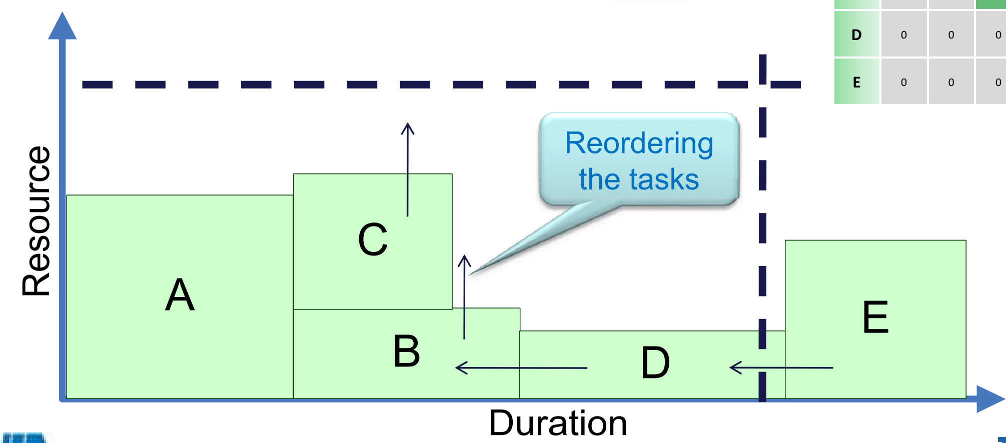
	A	B	C	D	E
A	**1**	0.9	0.7	0.3	0
B	0	**0.8**	0.4	0.6	0.25
C	0	0	**1**	0.5	0.5
D	0	0	0	**0.3**	1
E	0	0	0	0	**1**



Selecting the optimal solution



	A	B	C	D	E
A	**1**	0.9	0.7	0.3	0
B	0	**0.8**	0.4	0.6	0.25
C	0	0	**1**	0.5	0.5
D	0	0	0	**0.3**	1
E	0	0	0	0	**1**



Results – Using genetic algorithm

- 4 different kinds of resources
- 50% uncertain relations

Size of PEM matrices	Number of all possible solutions	Runtime of full evaluating algorithm	Runtime of genetic algorithm
10x10	2^{23}= 8,388,608	62 ms	42 ms
20x20	2^{95}	2.1 hours	580 ms
50x50	2^{613}	12.2 hours	83 sec

$2^{\frac{10*9}{2}} * 50\% = 2^{23}$

WBS-code				Tasks	Duration	Effort estimate (PersonDay)	Resource name
1. Sprint 4	1.2. Feature Enhancement GDS 2.1	1.1.1 Message Choreography	1.1.1.1	Regsitration Automation	5 days	5	YS
			1.1.1.2	Mapping	5 days	5	YS
			1.1.1.3	Export Interface Propagation	15 days	15	BC
			1.1.1.4	Process Flow Implementation - Response	15 days	15	PL
			1.1.1.5	Test, Review	5 days	5	AB
		1.2.1 Aut.Publ.	1.2.1.1	Core Implementation	5 days	5	AK
		1.2.2 Catalog in Flat File	1.2.2.1	UI	10 days	10	CK
			1.2.2.2	Core	10 days	10	GH
		1.2.3		Design for Search Publication Status	5 days	5	AB
	1.3. Pricing	1.3.1		Interface	5 days	5	YS
		1.3.2		Import Mapping , Import Module	5 days	5	CH
		1.3.3		UI Main	5 days	5	GH
		1.3.4		UI Price Component Detail	10 days	10	TK
		1.3.5		TOOLS Upgrade - Respository	10 days	10	AK
		1.3.6		Outgoing Process	15 days	15	AZ
		1.3.7		Authorization (Operation, Visibility) - DESIGN	5 days	5	TK
	1.4. Others	1.4.1		GDS 2.1 + GDS 2.0 Repository with Validation Update	7 days	7	RK
		1.4.2 EJB or WS DESIGN 2.1	1.4.2.1	Design	5 days	5	VS
			1.4.2.2	Implementation	5 days	5	LC
		1.4.3		ARIS Modeling	2 days	2	VS
		1.4.4		Installation Check	5 days	5	LC
		1.4.5		NW Log Completition	5 days	5	CH
		1.4.6		1Sync 6.5 Release analysis	3 days	3	RK
		1.4.7		GDS 2.1 Test, Bugfix	5 days	5	BH
		1.4.8		GDS 2.0 SP2 Test and Release Activities	5 days	5	BH
SP.1				Code Freeze	0 days	0	
SR.1				Sprint Review	0 days	0	

BY MODELLING DEPENDENCIES

WBS-code				Resource name	1.1.1.1	1.1.1.2	1.1.1.3	1.1.1.4	1.1.1.5	1.2.1.1	1.2.2.1	1.2.2.2	1.2.3	1.3.1	1.3.2	1.3.3	1.3.4	1.3.5	1.3.6	1.3.7	1.4.1	1.4.2.1	1.4.2.2	1.4.3	1.4.4	1.4.5	1.4.6	1.4.7	1.4.8	SP.1	SR.1
1. Sprint 4	1.2. Feature Enhancement GDS 2.1	1.1.1 Message Choreography	1.1.1.1	YS	1	1	0.9																								
			1.1.1.2	YS		1								1																	
			1.1.1.3	BC			1	1																							
			1.1.1.4	PL				1	1																						
			1.1.1.5	AB					1				1																	
		1.2.1 Aut.Publ.	1.2.1.1	AK						1	0.6	0.3						1													
		1.2.2 Catalog in Flat File	1.2.2.1	CK							1																				
			1.2.2.2	GH								1				1															
			1.2.3	AB									0.5																	
	1.3. Pricing		1.3.1	YS										1	1			0.4						0.5							
			1.3.2	CH											1	1										1					
			1.3.3	GH												1	1	0.8													
			1.3.4	TK													1		0.3	1											
			1.3.5	AK														1	0.7												
			1.3.6	AZ															1												
			1.3.7	TK																1											
	1.4. Others		1.4.1	RK																	1	0.9	0.3				1				
		1.4.2 EJB or WS DESIGN 2.1	1.4.2.1	VS																		0.8	1	1							
			1.4.2.2	LC																			1	0.5	1			0.5			
			1.4.3	VS																				1				0.5			
			1.4.4	LC																					1			1			
			1.4.5	CH																						0.6		0.5			
			1.4.6	RK																							1	0.5			
			1.4.7	BH																								0.7	1		
			1.4.8	BH																									1	1	
			SP.1																											1	1
			SR.1																												1

BY MODELLING DEPENDENCIES

Results of using genetic algorithm

- Constraints:
  - 15 people
  - 40 days
- Parameters:
  - Number of generation: 10
  - Size of the population: 75

Number of all possible solutions 2^{15}= **32768**.

Calculation with genetic algorithm needs 10*75 = **750** solutions.

2 objective functions: maximal probability and minimal lead time

3 objective functions: maximal probability, minimal lead time and maximal resource using

3 objective functions with weights

Objective functions	Run Time (ms)	Probability	Lead Time (day)	Resource (person)
P, LT	7444	0.510495	40	7
P, LT, R	7746	0.402011	40	9
40*P, 40*LT, 20*R	7190	0.510495	40	7

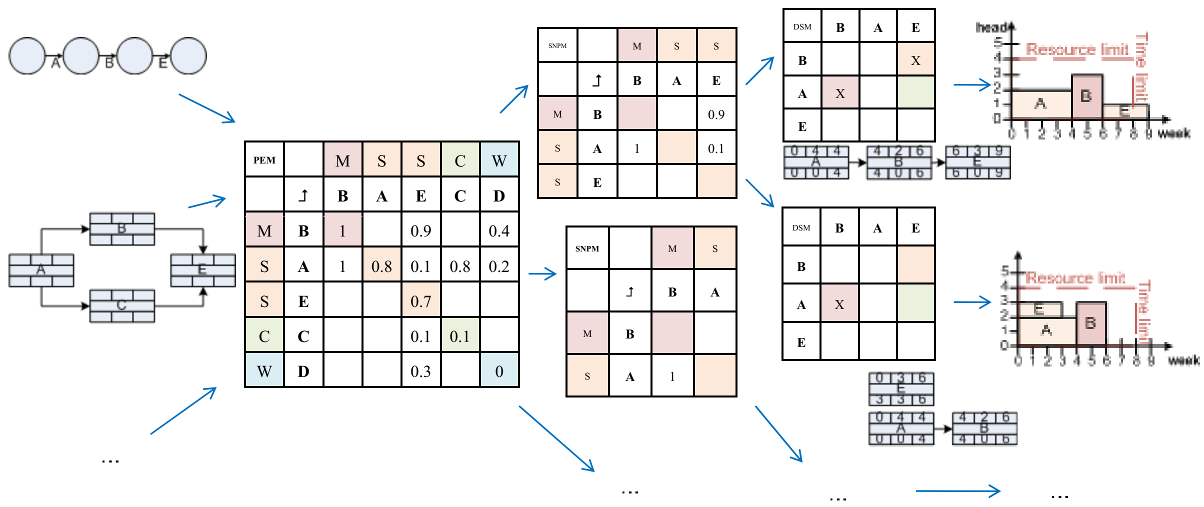
The solution according to multiple objective functions

WBS-code				Resource name	1.1.1.1	1.1.1.2	1.1.1.3	1.1.1.4	1.1.1.5	1.2.1.1	1.2.2.1	1.2.2.2	1.2.3	1.3.1	1.3.2	1.3.3	1.3.4	1.3.5	1.3.6	1.3.7	1.4.1	1.4.2.1	1.4.2.2	1.4.3	1.4.4	1.4.5	1.4.6	1.4.7	1.4.8	SP.1	SR.1
1. Sprint 4	1.2. Feature Enhancement GDS 2.1	1.1.1 Message Choreography	1.1.1.1	YS	1	1																									
1. Sprint 4	1.2. Feature Enhancement GDS 2.1	1.1.1 Message Choreography	1.1.1.2	YS		1								1																	
1. Sprint 4	1.2. Feature Enhancement GDS 2.1	1.1.1 Message Choreography	1.1.1.3	BC			1	1																							
1. Sprint 4	1.2. Feature Enhancement GDS 2.1	1.1.1 Message Choreography	1.1.1.4	PL				1	1																						
1. Sprint 4	1.2. Feature Enhancement GDS 2.1	1.1.1 Message Choreography	1.1.1.5	AB					1				1																		
1. Sprint 4	1.2. Feature Enhancement GDS 2.1	1.2.1 Aut.Publ.	1.2.1.1	AK						1	1	.						1													
1. Sprint 4	1.2. Feature Enhancement GDS 2.1	1.2.2 Catalog in Flat File	1.2.2.1	CK							1																				
1. Sprint 4	1.2. Feature Enhancement GDS 2.1	1.2.2 Catalog in Flat File	1.2.2.2	GH								1				1															
1. Sprint 4	1.2. Feature Enhancement GDS 2.1		1.2.3	AB									1																		
1. Sprint 4	1.3. Pricing		1.3.1	YS										1	1			.						.							
1. Sprint 4	1.3. Pricing		1.3.2	CH											1	1										1					
1. Sprint 4	1.3. Pricing		1.3.3	GH												1	1	.													
1. Sprint 4	1.3. Pricing		1.3.4	TK													1		.	1											
1. Sprint 4	1.3. Pricing		1.3.5	AK														1	.												
1. Sprint 4	1.3. Pricing		1.3.6	AZ															1												
1. Sprint 4	1.3. Pricing		1.3.7	TK																1											
1. Sprint 4	1.4. Others		1.4.1	RK																	1						1				
1. Sprint 4	1.4. Others	1.4.2 EJB or WS DESIGN 2.1	1.4.2.1	VS																		1	1	1							
1. Sprint 4	1.4. Others	1.4.2 EJB or WS DESIGN 2.1	1.4.2.2	LC																			1		1			.			
1. Sprint 4	1.4. Others		1.4.3	VS																				1				1			
1. Sprint 4	1.4. Others		1.4.4	LC																					1			1			
1. Sprint 4	1.4. Others		1.4.5	CH																						1		.			
1. Sprint 4	1.4. Others		1.4.6	RK																							1	1			
1. Sprint 4	1.4. Others		1.4.7	BH																								1	1		
1. Sprint 4	1.4. Others		1.4.8	BH																									1	1	
			SP.1																											1	1
			SR.1																												1

Summary

Planning and modelling methods

Prior project experience → PEM → SNPM → DSM/ network planning methods → Time, Cost and Resource planning



WHAT KIND OF prior project experience can be used?

WHAT can be realised from tasks?

HOW (in what kind of order) do these tasks have to be realised?

HOW MUCH time, cost and resource need for project realisation?