Presentation for ISCAS – 23/10/2008

Simulation-Based Planning, Re-Planning and Stability Analysis for Operational Release Plans

Dietmar Pfahl
Simula Research Laboratory & University of Oslo

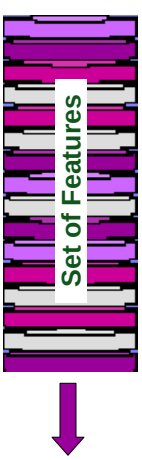
[simula . research laboratory]

UNIVERSITETET I OSLO

Outline

- Software Release Planning Problem
- Simulation-Based Operational Release Planning
- Planning and Dynamic Re-Planning
- Stability Analysis
- Work-in-Progress and Future Work
- Conclusion

Evolutionary Software Development



! Technological Constraints

✓ Shorter Time-to-Market

Project 1 → Release 1

! Resource Constraints

✓ Easier to Maintain

✓ Earlier Customer Feedback

Project 2 → Release 2

! Risks

! Budget Constraints

✓ Better Resource Allocation

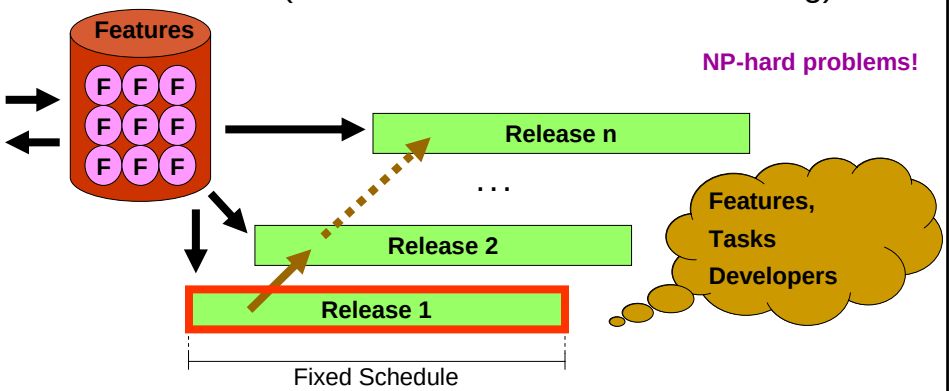
Project 3 → Release3

! Stakeholder Value

✓ flexible approach → better product → better customer satisfaction

Software Release Planning Problem

- Strategic Level → "sort of" Knapsack Problem
- Operational Level → "sort of" Project Planning Problem (→ resource allocation / scheduling)



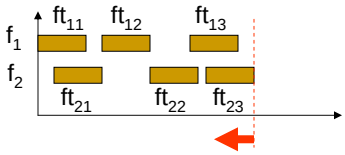
Operational Release Planning Problem

Consider this 7-tuple <F, T, FT, D, E, P, Dep>:

- F – set of release-specific features *f* [1, …, k]
- T – set of task types *t* [1, …, m]
- FT – set of feature/task type-combinations (= tasks) *ft* [1, …, km]
- D – set of developers *d* [1, …, n]
- E – set of estimated efforts *eff* per task [1, …, km]
- P – set of estimated (relative) productivities *prod* per task [1, …, nm]
- Dep – dependency relationships *dep* between subsequent task types [1, …, m-1]

Goal:

- assign developers d ∈ D to tasks ft ∈ FT such that



$max_{i=1...k}\{end\text{-}time(ft_{im})\} \rightarrow min$

(and additional restrictions hold, i.e., one-to-one mapping of d_k to ft_{ij})

Example with Start-Start Dependency

- **Features (F)**
  - **Estimated workload (effort) per task type [e.g., Person-Weeks (PW)]**
- **Task Types (T): design, implementation, test, etc.**
  - **dependency**
- **Developers (D)**
  - **(relative) productivity per task type [0 … 2]**
  - **one task at a time**
  - **no task change once assigned**

	F_1	F_2	…	…	…	F_k
T_1	4	3	…	…	…	2
…	…	…	…	…	…	…
T_m	5	7	…	…	…	9

?

	D_1	D_2	…	…	…	D_n
T_1	1	0.5	…	…	…	2
…	…	…	…	…	…	…
T_m	0.5	1.5	…	…	…	1

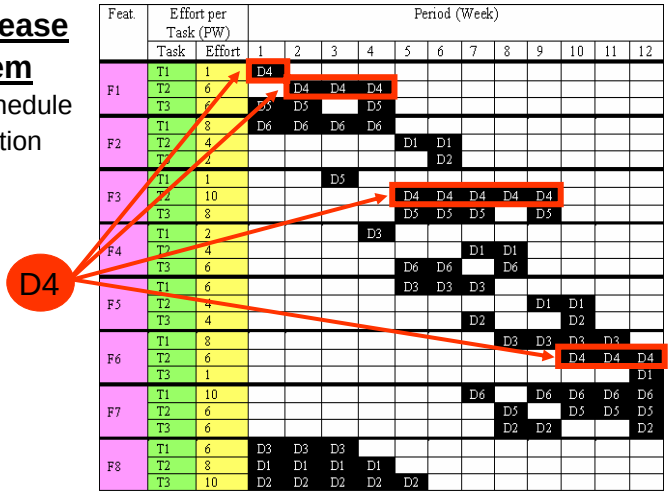
Example with Start-Start Dependency

Operational Release Planning Problem

- Feature/Task Schedule
- Developer Allocation

Example Case:

- 8 Features
- 3 Task Types
- 6 Developers



Feat.	Task	Effort	1	2	3	4	5	6	7	8	9	10	11	12
		Effort per Task (PW)	Period (Week)											
F1	T1	1	D4											
	T2	6		D4	D4	D4								
	T3	6	D5	D5		D5								
F2	T1	8	D6	D6	D6	D6								
	T2	4					D1	D1						
	T3	2						D2						
F3	T1	1			D5									
	T2	10					D4	D4	D4	D4	D4			
	T3	8					D5	D5	D5		D5			
F4	T1	2				D3								
	T2	4							D1	D1				
	T3	6					D6	D6		D6				
F5	T1	6					D3	D3	D3					
	T2	4									D1	D1		
	T3	4							D2			D2		
F6	T1	8								D3	D3	D3	D3	
	T2	6										D4	D4	D4
	T3	1												D1
F7	T1	10							D6		D6	D6	D6	D6
	T2	6								D5		D5	D5	D5
	T3	6								D2	D2			D2
F8	T1	6	D3	D3	D3									
	T2	8	D1	D1	D1	D1								
	T3	10	D2	D2	D2	D2	D2							

[NgR05] Ngo-The, A and Ruhe, G (2005) Optimized Resource Allocation for Incremental Software Development (submitted to Transactions on SE).

Outline

- **Software Release Planning Problem**
- **Simulation-Based Operational Release Planning**
- **Planning and Dynamic Re-Planning**
- **Stability Analysis**
- **Work-in-Progress and Future Work**
- **Conclusion**

Operational Release Planning (ORP)

- Example:

Allocate developers to tasks such that:

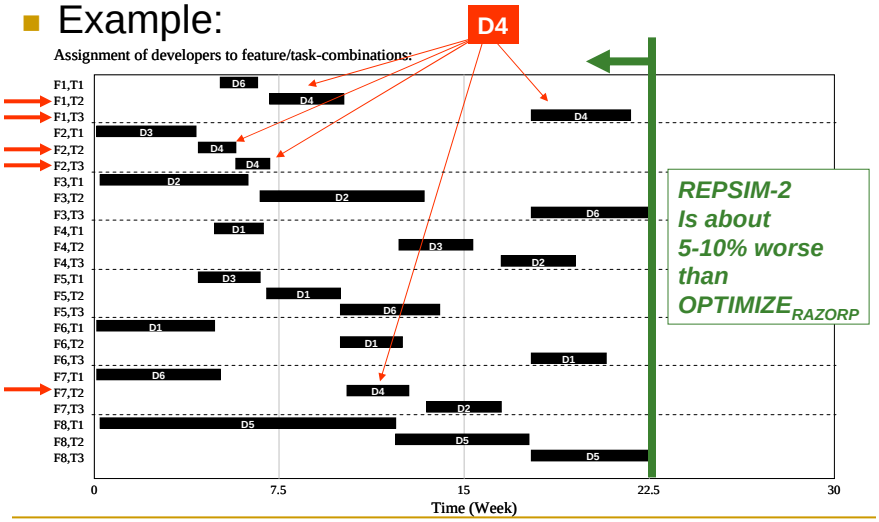
$\max_{i=1...8}\{\text{end-time}(task_{i,3})\} \rightarrow \min$

m ↓ / k Features →		Effort Estimates [person-week]								Productivity Estimates [dimensionless]					
		F1	F2	F3	F4	F5	F6	F7	F8	D1	D2	D3	D4	D5	D6
Task Type	T1: Design	3	8	6	3	5	7	10	6	1.5	1	2	0	0.5	2
	T2: Implementation	6	3	10	3	6	5	5	8	2	1.5	1	2	1.5	1
	T3: Test	6	2	5	6	4	3	6	10	1	2	0	1.5	2	1

- End-start dependency between subsequent tasks
- 1-to-1 mapping between developers and tasks

So why *simulation-based* ORP ?

- Example:



Simulation-Based ORP

- **Why?**
  - **More flexibility with regards to**
    - dynamic re-planning,
    - stability analyses, and
    - problem refinement

    as compared to (static) optimization algorithms
- **How?**
  - **Using either discrete-event or continuous process simulation models (have both)**
- **What?**
  - **(Planning and) Re-planning**
  - **Stability Analysis**

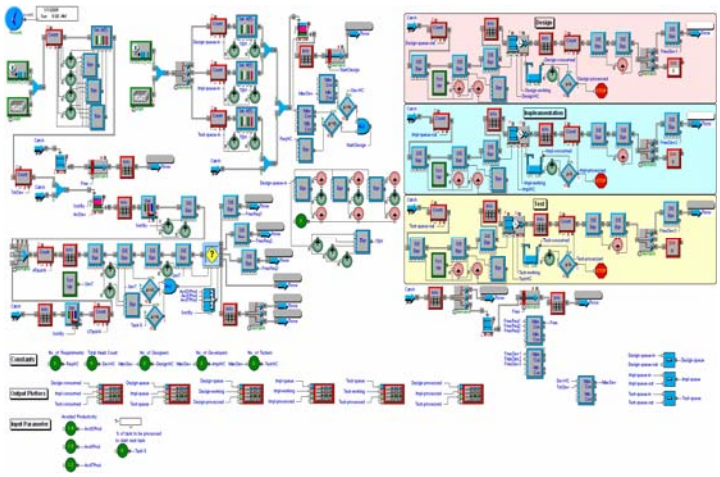
Outline

- **Software Release Planning Problem**
- **Simulation-Based Operational Release Planning**
- **Planning and Dynamic Re-Planning**
- **Stability Analysis**
- **Work-in-Progress and Future Work**
- **Conclusion**

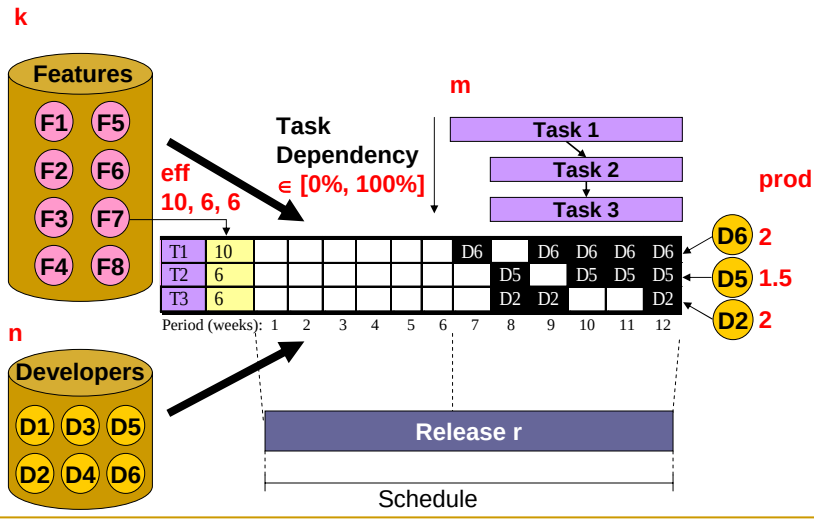
Planning – Simulation Model DynaReP

Technique:
Discrete-Event

Tool:
EXTEND®



Planning – Model Parameters



Planning – Heuristic

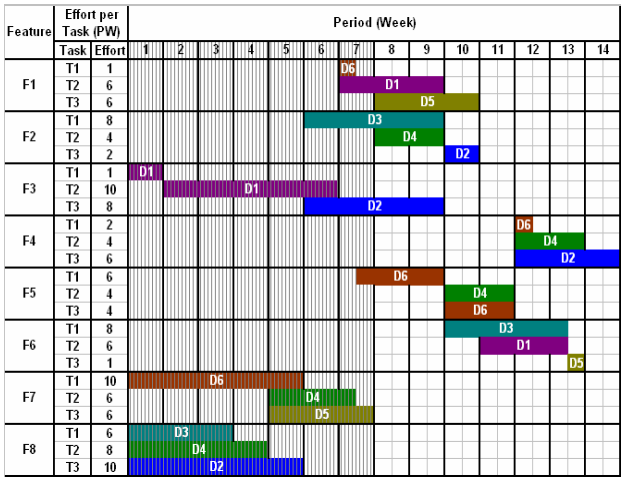
- Assign the next available developer with the highest task-specific productivity to the next waiting feature with the largest effort (for a specific task).

Note:

- If only developers with very low productivity are currently idle, this mapping rule can result in assigning a developer with low productivity to a large feature → will become a bottleneck!
- To avoid such a worst case situation, threshold variables are defined which exclude developers with a productivity that is too low.
- Finding a good set of threshold values can be automized (using a built-in optimization functionality)

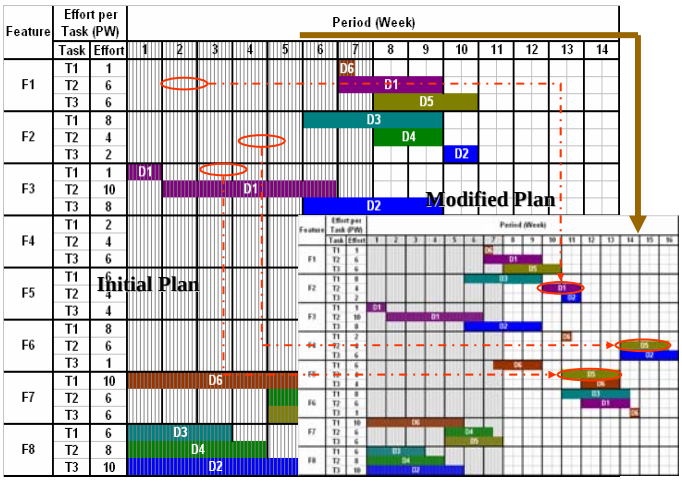
Planning Re-Planning

Initial Plan



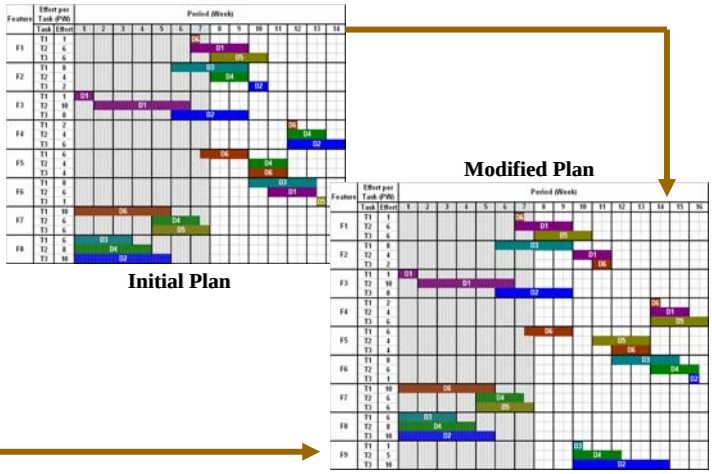
Dynamic Re-Planning (Example 1)

Developer D4 becomes unavailable after end of the 7th week



Dynamic Re-Planning (Example 2)

Feature F9 is added after end of the 7th week



Re-Planning – Other Possible Analyses

- Drop in/out of developers
- Addition/deletion of features
- Overestimated or underestimated efforts
- Overestimated or underestimated productivities
- Varying task type dependencies

All the above:

- In any combination
- At any point in time (repeatedly)

Re-Planning – Additional Analyses after Model Enhancement

In addition or complimentary to the previous:

- Productivities defined per feature (not task type)
- Feature dependencies

Other aspects:

- Learning effects during development
- Time pressure effects
- … and many others

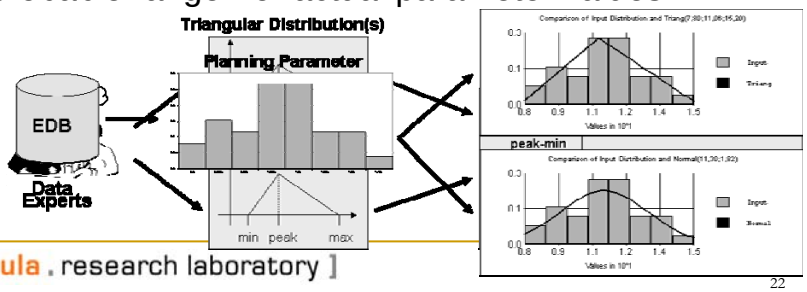
Outline

- Software Release Planning Problem
- Simulation-Based Operational Release Planning
- Planning and Dynamic Re-Planning
- Stability Analysis
- Work-in-Progress and Future Work
- Conclusion

Stability Analysis – Why?

Problem:

- Planning parameters are estimates
  - Based on empirical data
  - Based on expert knowledge
- It makes sense to assume distributions that define a "probable range" for actual parameter values



Stability Analysis

Main question in the following:

- How sensitive does the initial operational plan react to changes in any of the planning parameters, i.e.
  - Effort estimates
  - Productivity estimates
  - Task type dependencies

Two classes of analyses currently possible:

- Developer allocations to tasks are
  - Fixed → Analyze effect on work-backlog
  - Flexible → Analyze effect on duration and plan structure

Stability Analysis

Main Question:

- How sensitive does the initial operational plan react to changes in any of the planning parameters, i.e.
  - Effort estimates
  - Productivity estimates
  - Task type dependencies

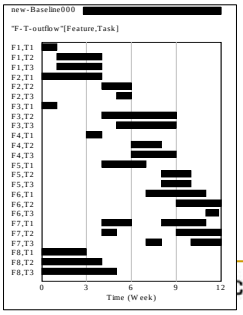
Two classes of analyses currently possible:

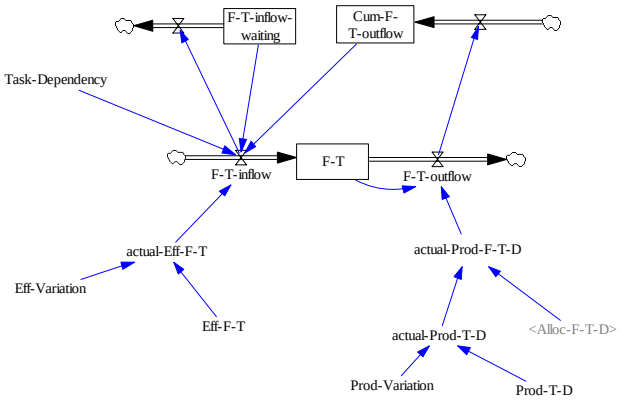
- Developer allocations to tasks are
  - Fixed → Analyze effect on work-backlog
  - Flexible → Analyze effect on duration and plan structure

Stability Analysis – REPSIM-1

Technique:
System Dynamics

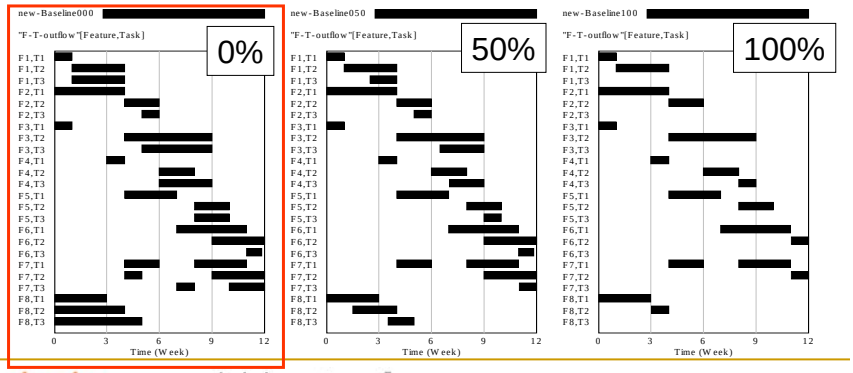
Tool:
Vensim®



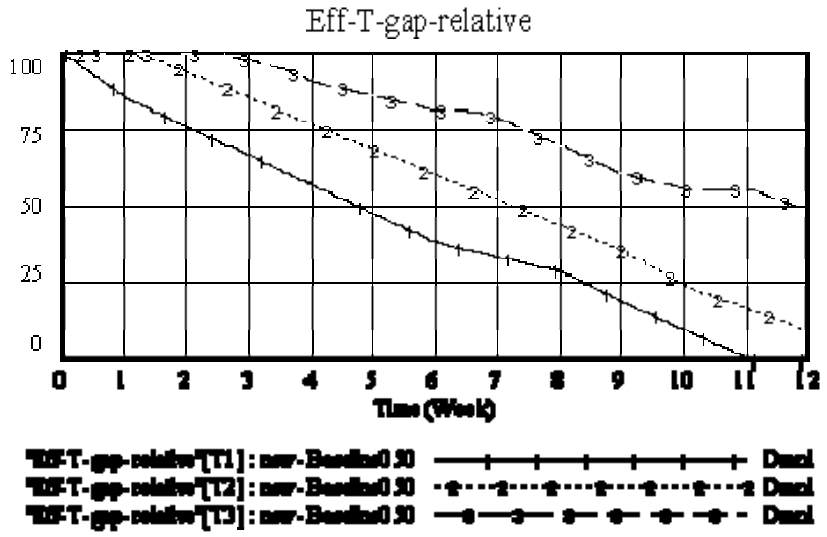


Stability Analysis – Example 1

- Gantt charts showing active tasks for varying task dependencies (Baseline = Task Type Dependency 0%)

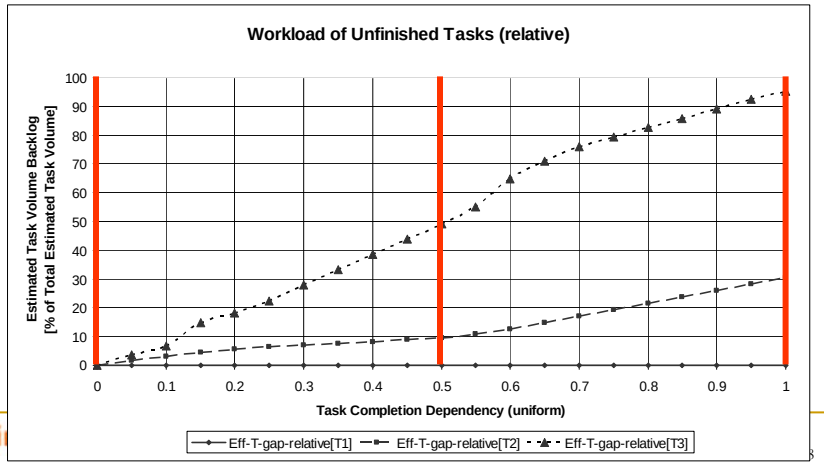


Stability Analysis – Example 1



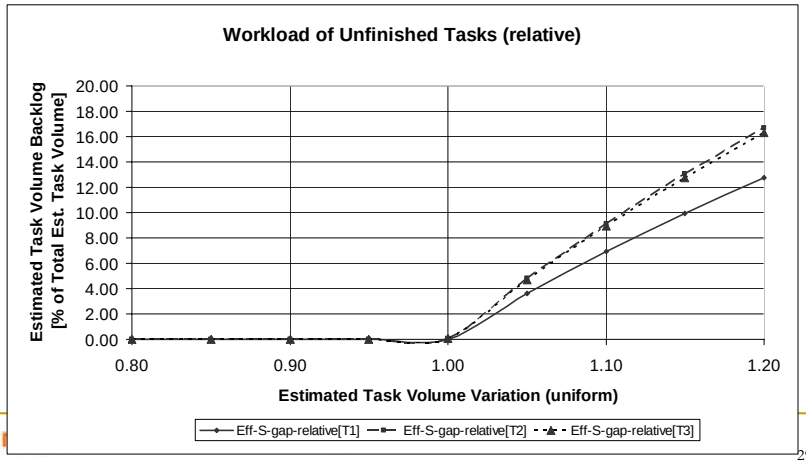
Stability Analysis – Example 1

- Task type-specific work backlog (cumulated over features) for varying task dependencies



Stability Analysis – Example 2

- Task type-specific work backlog (cumulated over features) for varying effort estimates

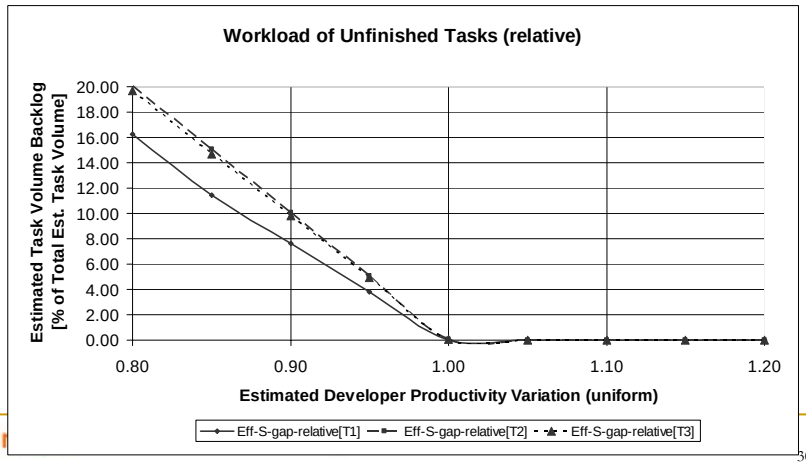


[si

29

Stability Analysis – Example 3

- Task type-specific work backlog (cumulated over features) for varying productivity estim.

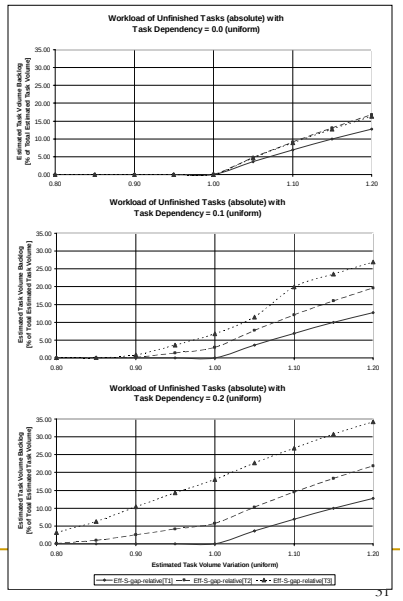


[si

30

Stability Analysis – Example 1+2

- Task type-specific work backlog (cumulated over features) for varying task dependencies and varying effort estimates



Stability Analysis

Main Question:

- How sensitive does the initial operational plan react to changes in any of the planning parameters, i.e.
  - Effort estimates
  - Productivity estimates
  - Task type dependencies

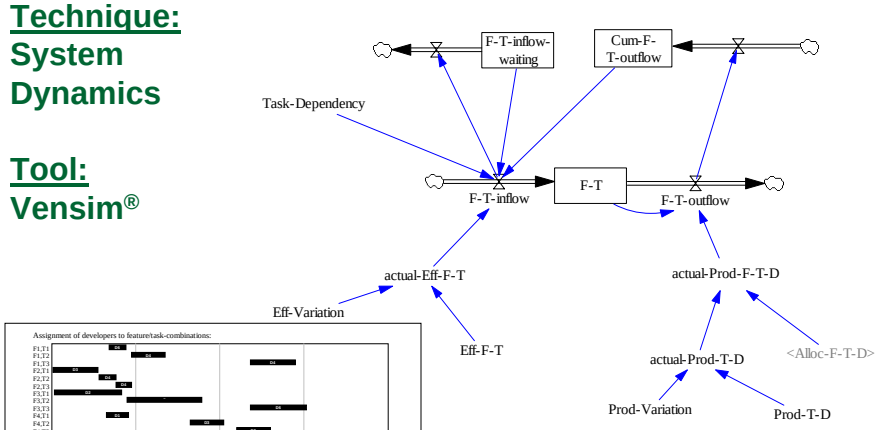
Two classes of analyses currently possible:

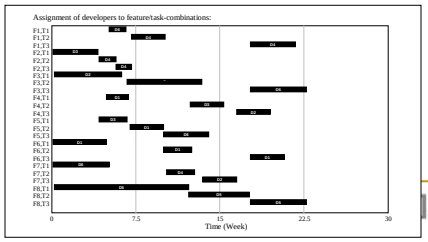
- Developer allocations to tasks are
  - Fixed → Analyze effect on work-backlog
  - Flexible → Analyze effect on duration and plan structure

Stability Analysis – REPSIM-2

Technique:
System Dynamics

Tool:
Vensim®



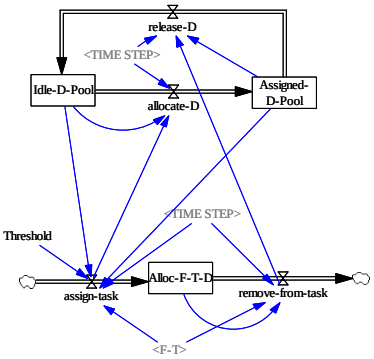


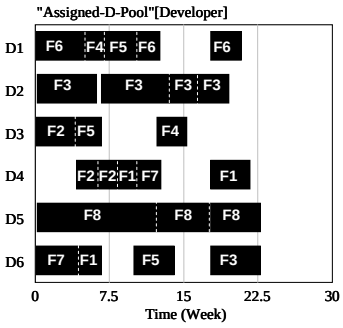
33

Stability Analysis – REPSIM-2

Technique:
System Dynamics

Tool:
Vensim®





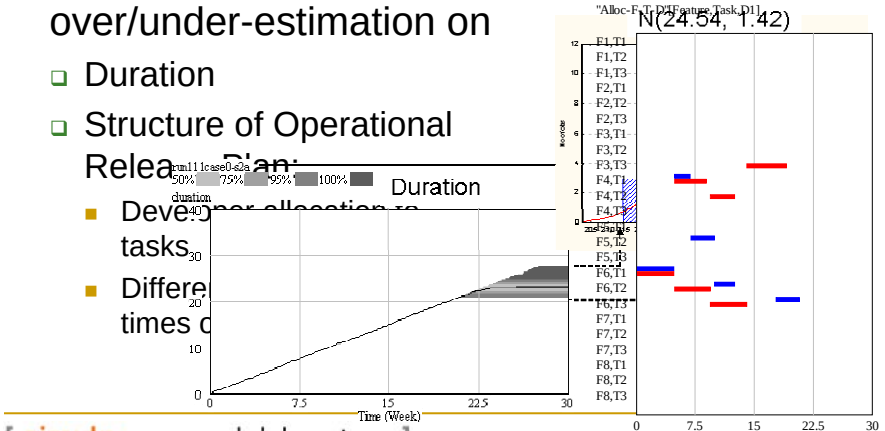
		Effort Estimates [Person-Week]								Productivity Estimates [Dimensionless]					
		F1	F2	F3	F4	F5	F6	F7	F8	D1	D2	D3	D4	D5	D6
Task Type	T1: Design	3	8	6	3	5	7	10	6	1.5	1	2	0	0.5	2
	T2: Implementation	6	3	10	3	6	5	5	8	2	1.5	1	2	1.5	1
	T3: Test	6	2	5	6	4	3	6	10	1	2	0	1.5	2	1

[simula . res

Stability Analysis – Research Goal

- Research Question:

  Analyze impact of effort and productivity over/under-estimation on
  - Duration
  - Structure of Operational Release Plan:
    - Developer allocation to tasks
    - Differences in start and end times of tasks



Stability Analysis – Hypotheses

- Decrease (increase) of productivity and increase (decrease) of effort results in …
  - H1: significant increase (decrease) of release duration.
  - H2: significant instability in the assignment of developers to tasks.
  - H3: significant instability in the start and end times of tasks.

Stability Analysis – Experiments

- Baseline Case
- Simulations

Table 1a. Feature/Task-specific efforts

		eff(i, j) [person-week]							
		f(1)	f(2)	f(3)	f(4)	f(5)	f(6)	f(7)	f(8
Task	t(1)	3	8	6	3	5	7	10	6
	t(2)	6	3	10	3	6	5	5	8
	t(3)	6	2	5	6	4	3	6	10

Table 1b. Developer-specific productivities

		p(k, j) [no unit]					
		d(1)	d(2)	d(3)	d(4)	d(5)	d(6)
Task	t(1)	1.5	1	2	0	0.5	2
	t(2)	2	1.5	1	2	1.5	1
	t(3)	1	2	0	1.5	2	1

Table 2. ProSim/ORP risk factor variation

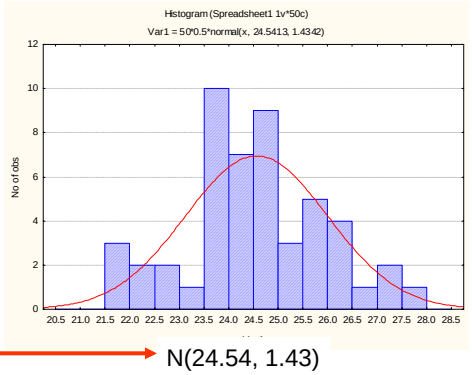
Case	Variation Parameter	Min	Max
Case 1	Effort	-10%	20%
Case 2	Effort	-20%	20%
Case 3	Effort	-20%	40%
Case 4	Effort	-40%	40%
Case 5	Productivity	-20%	10%
Case 6	Productivity	-20%	20%
Case 7	Productivity	-40%	20%
Case 8	Productivity	-40%	40%
Case 9	Mixed	Case 1 + Case 5	
Case 10	Mixed	Case 2 + Case 6	
Case 11	Mixed	Case 3 + Case 7	
Case 12	Mixed	Case 4 + Case 8	

Stability Analysis – Results

- Effect on Duration
- Distribution Fitting

Table 3a. Summary statistics of ORP performance parameter Dur_run

Case	Baseline Duration [weeks]	Run-specific duration Dur_run [weeks]	
		Mean	Standard dev.
Case 1	22.71875	23.811	1.611
Case 2	22.71875	23.146	1.608
Case 3	22.71875	24.755	2.210
Case 4	22.71875	23.526	2.428
Case 5	22.71875	23.780	1.012
Case 6	22.71875	23.075	1.250
Case 7	22.71875	24.849	2.034
Case 8	22.71875	23.508	2.141
Case 9	22.71875	24.541	1.434
Case 10	22.71875	23.038	2.080
Case 11	22.71875	26.367	2.495
Case 12	22.71875	23.950	3.331



N(24.54, 1.43)

Stability Analysis – Results

- Effect on Duration

Case (50 runs each)	Deviation of Effort Mean from Baseline Effort	Deviation of Productivity Mean from Baseline Productivity	Deviation of Duration Mean from Baseline Duration
Case 1	3.4%	0%	4.8%
Case 2	0%	0%	1.9%
Case 3	6.8%	0%	9.0%
Case 4	0%	0%	3.6%
Case 5	0%	-3.3%	4.7%
Case 6	0%	0%	1.6%
Case 7	0%	-6.7%	9.4%
Case 8	0%	0%	3.5%
Case 9	3.4%	-3.7%	8.0%
Case 10	0%	0%	1.4%
Case 11	6.9%	-7.5%	16.1%
Case 12	0%	-0.1%	5.4%

- Single Sample T-Test

Table 5. Results from the single sample t-tests comparing the distribution of the average durations in each case with the duration in the baseline case

Case (n=50)	Baseline Duration	Duration Mean	Duration Std. Dev.	p-Value	
Case 1	22.71875	23.811	1.611	0.000	
Case 2	22.71875	23.146	1.608	0.066	?
Case 3	22.71875	24.755	2.210	0.000	
Case 4	22.71875	23.526	2.428	0.023	
Case 5	22.71875	23.780	1.012	0.000	
Case 6	22.71875	23.075	1.250	0.049	
Case 7	22.71875	24.849	2.034	0.000	
Case 8	22.71875	23.508	2.141	0.012	
Case 9	22.71875	24.541	1.434	0.000	
Case 10	22.71875	23.038	2.080	0.284	?
Case 11	22.71875	26.367	2.495	0.000	
Case 12	22.71875	23.950	3.331	0.012	

alpha = 0.05

Stability Analysis – Results

- Effect on Duration (with Effect Sizes)

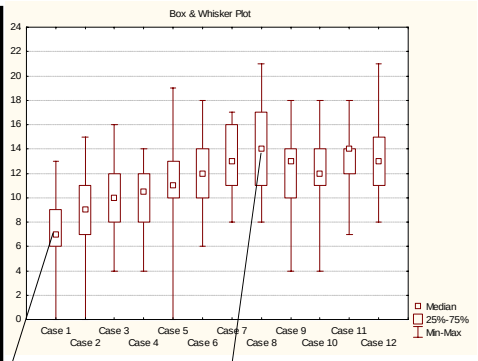
Case (n = 50)	Duration Baseline	Duration Mean	Duration Std. Dev.	Duration Difference	Duration Effect Size
Case 1	22.71875	23.811	1.611	1.092	0.678
Case 2	22.71875	23.146	1.608	0.427	0.266
Case 3	22.71875	24.755	2.21	2.036	0.921
Case 4	22.71875	23.526	2.428	0.807	0.332
Case 5	22.71875	23.78	1.012	1.061	1.049
Case 6	22.71875	23.075	1.25	0.356	0.285
Case 7	22.71875	24.849	2.034	2.130	1.047
Case 8	22.71875	23.508	2.141	0.789	0.369
Case 9	22.71875	24.541	1.434	1.822	1.271
Case 10	22.71875	23.038	2.08	0.319	0.153
Case 11	22.71875	26.367	2.495	3.648	1.462
Case 12	22.71875	23.95	3.331	1.231	0.370

Stability Analysis – Results

- Effect on Developer Allocation
- Total number of allocations = 8 x 3 = 24
- With 6 developers:

Table 3b. Summary statistics of ORP performance parameter Alloc_diff

Case	Difference in developer allocation Alloc_diff [no unit]	
	Mean	Standard dev.
Case 1	7.2	3.289
Case 2	9.04	3.194
Case 3	9.765	2.688
Case 4	10.14	2.298
Case 5	11.38	3.562
Case 6	12.26	2.776
Case 7	13.3	2.644
Case 8	14.06	3.191
Case 9	11.96	3.201
Case 10	12.16	2.78
Case 11	13.08	2.806
Case 12	13.3	2.652



Median: 29% change Median: 58% change

Stability Analysis – Results

- Effect on Task Scheduling
- Formulas:

$$Dv\_diff_i = \frac{d_{i,1} + d_{i,2}}{d_{i,1} + d_{i,2} + d_{i,3}}, \text{ with}$$

$$d_{i,1} = \sqrt{\sum_{j=1}^{24} (ST_j^i - ST_j^{baseline})^2},$$

$$d_{i,2} = \sqrt{\sum_{j=1}^{24} (ET_j^i - ET_j^{baseline})^2},$$

$$d_{i,3} = \sqrt{\sum_{j=1}^{24} (overlap_j^i)^2},$$

- Data:

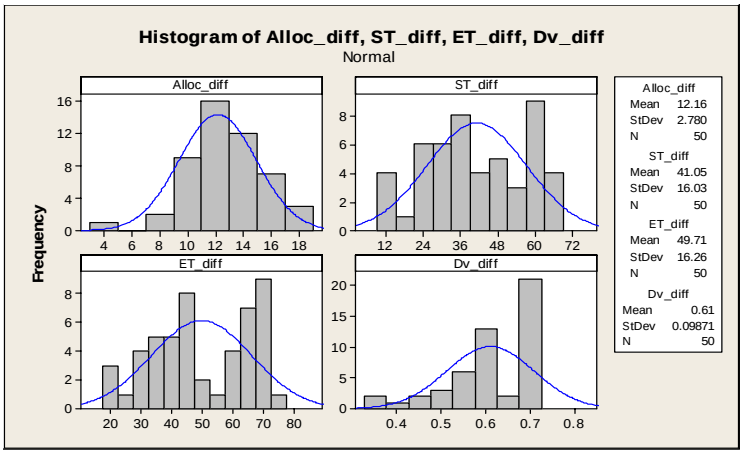
Table 3c. Summary statistics of ORP performance parameters ST_diff, ET_diff, Dv_diff

Case	ST_diff		ET_diff		Dv_diff	
	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.
Case 1	19.26	11.34	25.44	11.9	0.4214	0.1503
Case 2	27.67	16.67	35.28	17.79	0.5026	0.1533
Case 3	32.72	17.02	42.93	17.76	0.5345	0.136
Case 4	40.76	17.8	52.43	19.52	0.6002	0.1299
Case 5	28.15	17.8	33.71	17.84	0.4958	0.1465
Case 6	34.28	17.55	40.8	17.48	0.5536	0.1344
Case 7	46.01	20.96	54.09	21.32	0.6096	0.1306
Case 8	47.95	20.49	57.06	21.55	0.6368	0.112
Case 9	38.29	18.62	47.41	19.19	0.5742	0.123
Case 10	41.05	16.03	49.71	16.26	0.61	0.0987
Case 11	60.34	22.13	75.17	26.19	0.6902	0.0817
Case 12	62.53	32.56	75.81	30.18	0.7052	0.0717

Dv_diff ∈ [0, 1]

Stability Analysis – Results

- Details of Case 10

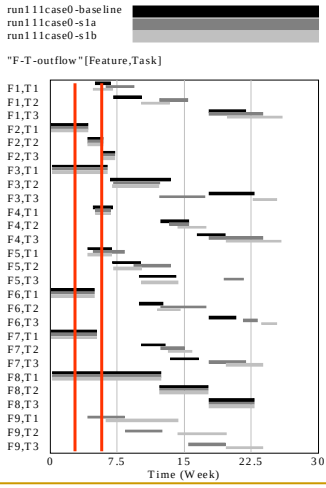


Stability Analysis – Summary

- H1: Variation Impact on Duration → ?
  - Asymmetric Variation: duration is always significantly different
  - Symmetric Variation: Cases 2 and 10 (and almost 8) have no significantly difference in duration (and effect size < 0.5, i.e., no "practical significance" according to Cohen)
- H2: Variation Impact on Developer Allocation → yes
  - Average change of developer allocation is in the range of 29-58% of all allocations
- H3: Variation Impact on Task Scheduling → yes
  - Average change of feature/task scheduling is in the range of 0.42-0.71 (where 1 is equivalent to zero overlap).
- Note:
  - These results have been corroborated in simulations with much larger releases (>60 features; up to 13 developers).

Additional Stability Analyses

- Feature inclusion (a: 3 weeks / b: 6 weeks)



Feature		Baseline						Scenario 1a (F9-3Wk)						Scenario 1b (F9-6Wk)					
	Task	Developer						Developer						Developer					
		D1	D2	D3	D4	D5	D6	D1	D2	D3	D4	D5	D6	D1	D2	D3	D4	D5	D6
F1	T1																		
	T2																		
	T3																		
F2	T1																		
	T2																		
	T3																		
F3	T1																		
	T2																		
	T3																		
F4	T1																		
	T2																		
	T3																		
F5	T1																		
	T2																		
	T3																		
F6	T1																		
	T2																		
	T3																		
F7	T1																		
	T2																		
	T3																		
F8	T1																		
	T2																		
	T3																		
F9	T1																		
	T2																		
	T3																		
Duration		22.71 Weeks						23.75 Weeks						25.78 Weeks					

Outline

- Software Release Planning Problem
- Simulation-Based Operational Release Planning
- Planning and Dynamic Re-Planning
- Stability Analysis
- Work-in-Progress and Future Work
- Conclusion

Work in Progress

- Model Enhancements
  - Facilitate more types of analyses
    - e.g., feature-dependency, (partly) fixed developer allocations in re-planning
  - Relax assumptions
    - e.g., 1-to-1 relationships between task types, 1-to-1 assignment of developers to tasks
- Complement with Empirical Studies

Conclusion

- Simulation-based planning/re-planning of releases on operational level may support decision makers in dynamic environments
- Simulation-based stability (or uncertainty) analysis could be an input to risk management
- Limitations:
  - Initial plans are not optimal (about 5-10%)
  - Experimental basis still small
  - “Risk-drivers” (internal properties) of ORPs not yet fully clear

Future Work

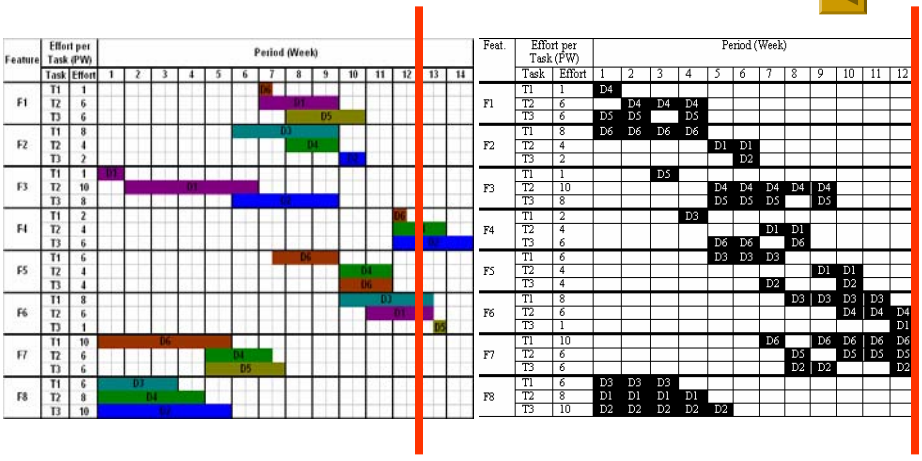
- Analyze properties of ORPs to better understand when duration is significantly effected by estimate uncertainty
- Improve heuristics or find way to integrate with (static) optimization algorithms
- Integrate with strategic product management (multi-release perspective)

References

- Al-Emran, A.; Khosrovian, K.; Pfahl, D.; Ruhe, G.: Studying the Impact of Uncertainty in Operational Release Planning. Submitted to ESEM 2007.
- Al-Emran, A.; Pfahl, D.: Operational Planning, Re-Planning and Risk Analysis for Software Releases. Accepted for publication in Proceedings of PROFES 2007.
- Al-Emran, A.; Pfahl, D.; Ruhe, G.: DynaReP – A Discrete Event Simulation Model for Re-planning of Software Releases. Accepted for publication in Proceedings of ICSP 2007.
- Pfahl, D.: ProSim/RA – Software Process Simulation in Support of Risk Assessment. In: S. Biffl, A. Aurum, B. Boehm, H. Erdogmus, P. Grünbacher (eds.), Value-based Software Engineering, Berlin: Springer, 2005, 263-286, ISBN 3-540-25993-7.
- Pfahl, D., Al-Emran, A., Ruhe, G.: Simulation-Based Stability Analysis for Software Release Plans. In: Wang, Q. et al. (eds.): SPW/ProSim 2006 - Proceedings. LNCS 3966, Berlin-Heidelberg: Springer, 2006, 262-273.
- Pfahl, D.; Al-Emran, A.; Ruhe, G.: A System Dynamics Model for Analyzing the Stability of Software Release Plans. In: Software Process Improvement and Practice (in print).

- Ngo-The, A., Ruhe, G.: Optimized Resource Allocation for Incremental Software Development. TR 062/2006, Laboratory for Software Engineering Decision Support, University of Calgary (2006)

Simulated Plan vs. Optimal Plan [NgR05]



Stability Analysis – Sim. Model B-2