11th CREST Open Workshop: SBSE for Early Lifecycle Software Engineering: Search Based Requirements and Project Management Optimisation

OPTIMIZATION METHODS FOR SOFTWARE RELEASE PLANNING

Guenther Ruhe
University of Calgary
Canada

Software Engineering Decision Support Laboratory

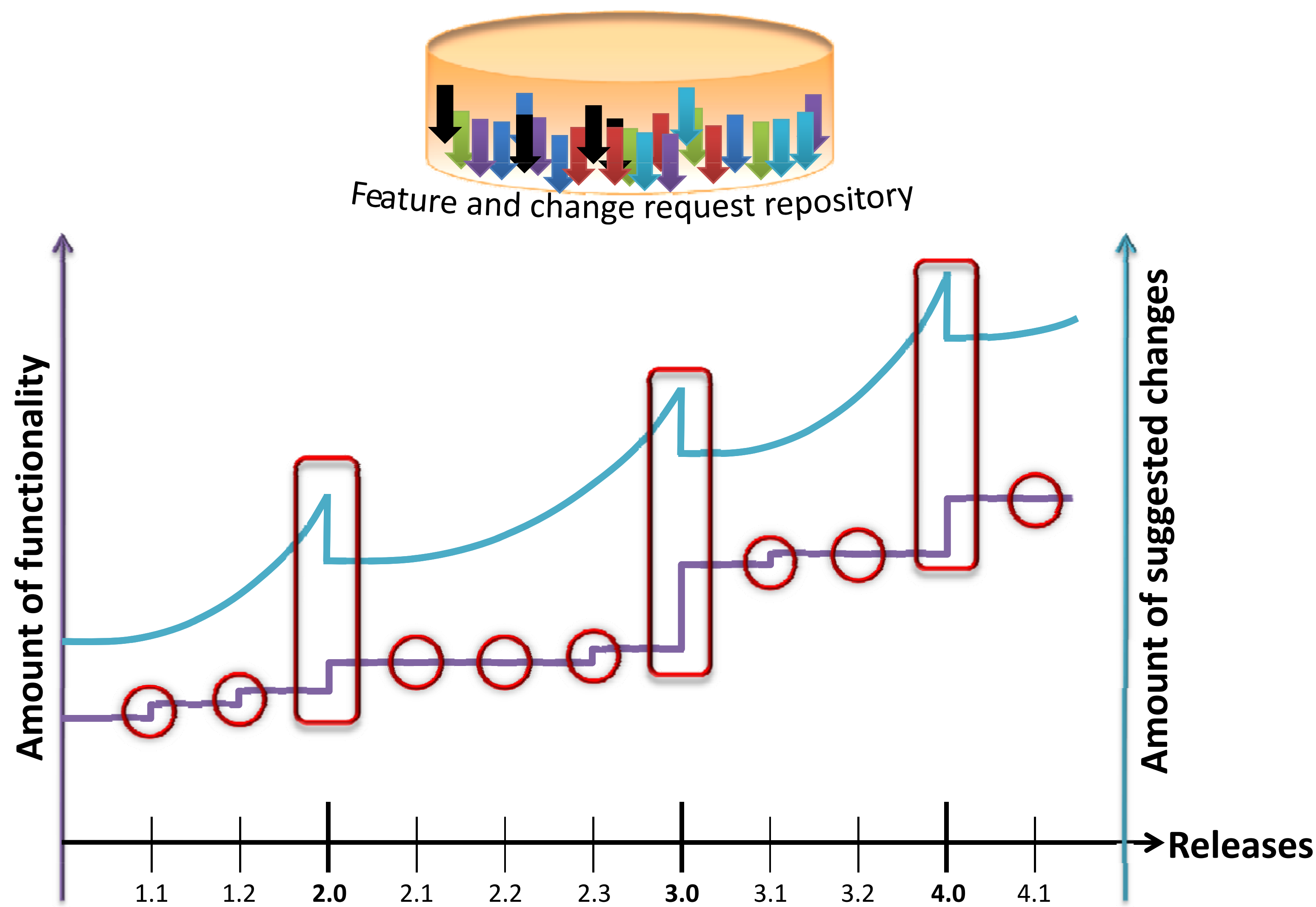
- Created in July 2001 at the University of Calgary
- Research team of 13 researchers (2 undergraduate, 3 graduate, 6 PhD students, and 2 profs)
- Research topics: Decision support (systems) for
  - Software release planning
  - Project management
  - Staffing
  - Effort estimation
  - COTS selection

- Research approach:
  - Interdisciplinary
  - Both fundamental and applied research
  - Empirical validation of results
- University spin-off company: Expert Decisions

Agenda

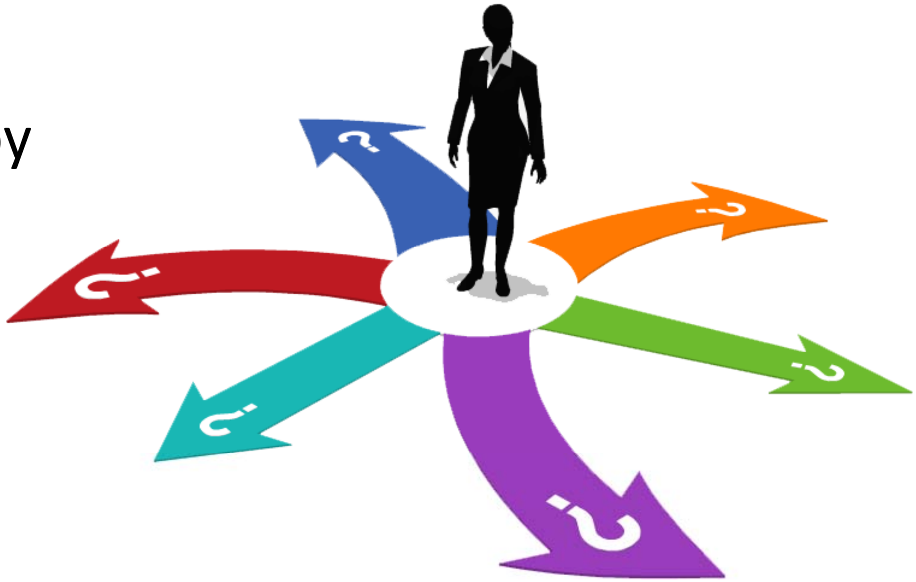
1. Decisions in release planning
2. Strategic release planning: Randomized versus deterministic
3. Operational release planning: Deterministic AND randomized
4. When-to-release decisions
5. Ongoing research

Release planning – What it is?

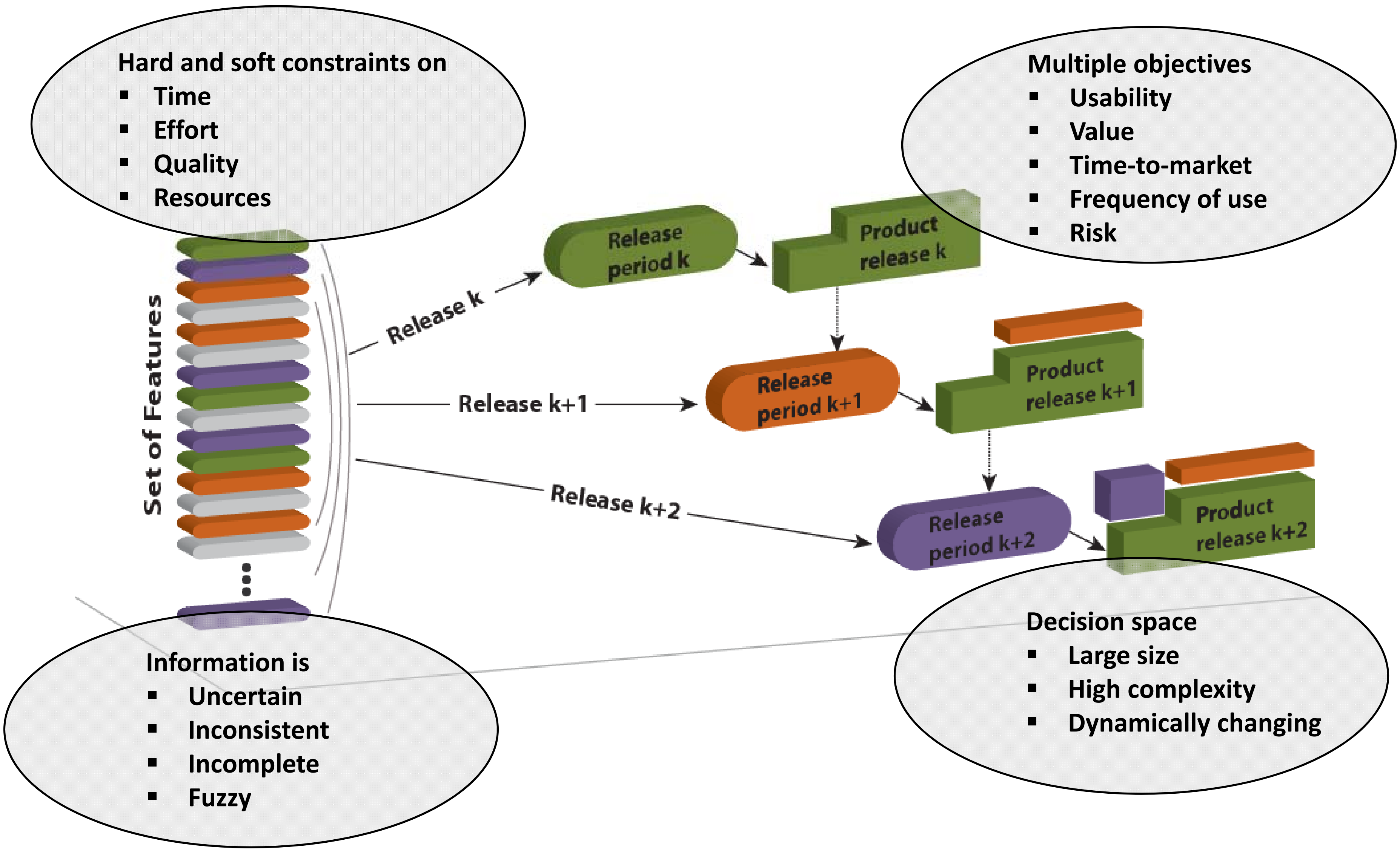


Release decisions

- Which features should be offered in the next release(s)?
- Which features should not be offered in the next release(s)?
- When is the best time for a product release?
- How to adjust to change for a given release?
  - When to re-plan?
  - How much to re-plan?
  - Which formerly planned features should be replaced by new ones?
  - How often re-planning can be done?



Release planning - Why efficient search is needed?



Agenda

1. Decisions in release planning
2. Strategic release planning: Randomized versus deterministic
3. Operational release planning: Deterministic AND randomized
4. When-to-release decisions
5. Ongoing research

“The mere formulation of a problem is far more essential than its solution, which may be merely a matter of mathematical or experimental skills. To raise new questions (and), new possibilities, to regard old problems from a new angle, requires creative imagination and marks real advances in science.”

(Albert Einstein, 1879-1955)

Optimized release planning – How it began

EVOLVE: Greer, D. and Ruhe, G., Software Release Planning: An Evolutionary and Iterative Approach, Information and Software Technology, Vol. 46 (2004), pp. 243-253.

What constitutes a release plan?

$$\text{Max}\{ F(x, \alpha) = (\alpha - 1)\ F1(x) + \alpha\ F2(x) \text{ subject to } 0 \leq \alpha \leq 1,\ x \text{ from } X\}$$

Stakeholders

Weightings for stakeholders

Scores of stakeholders towards urgency (F1) and value (F2)

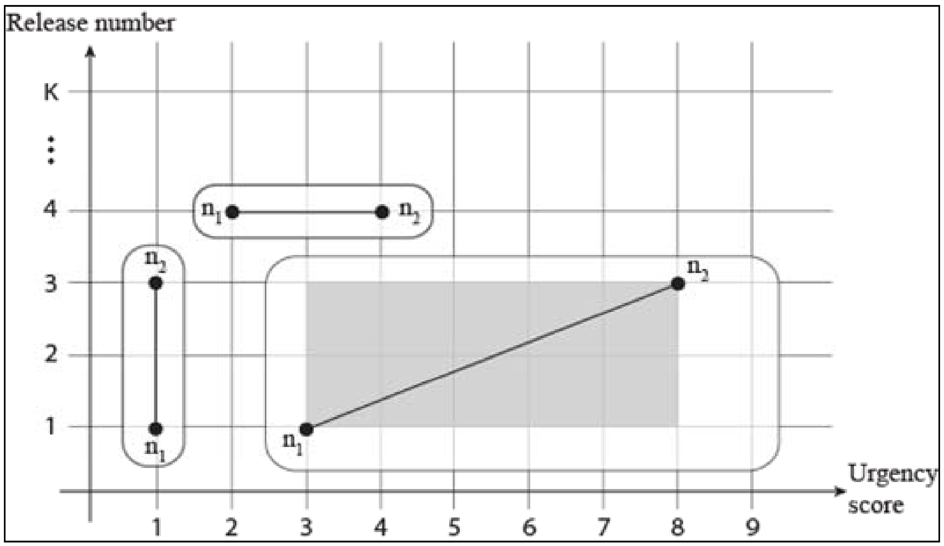
X composed of

- effort constraints

- coupling and precedence constraints (between features)

Optimized release planning – How it began

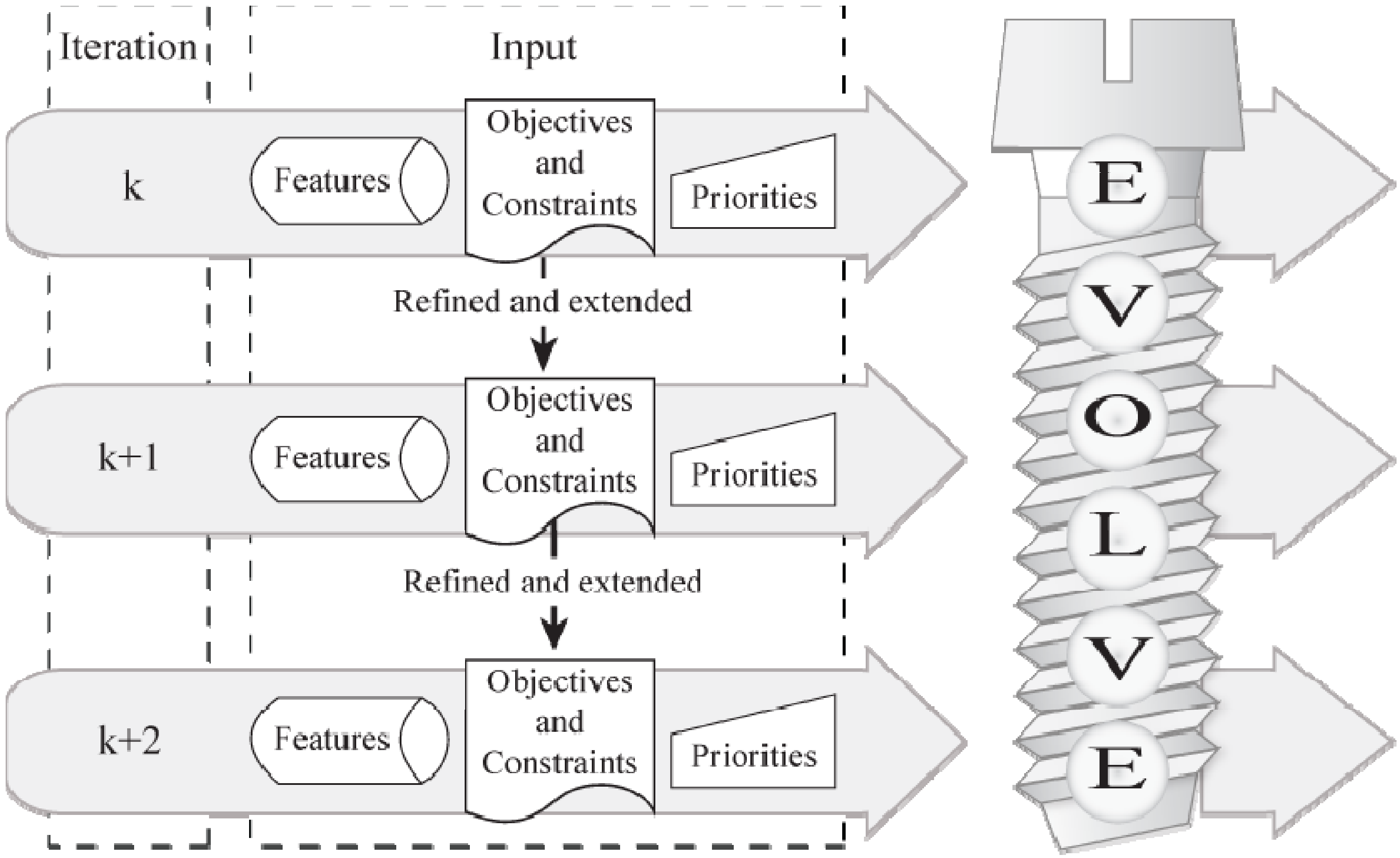
F1(x) is a penalty function defined for plan x describing the degree of violation of the monotonicy property between all pairs of features



F2(x) is a benefit function based on feature scores of the stakeholders and the actual assignment of the feature according to the plan under consideration.

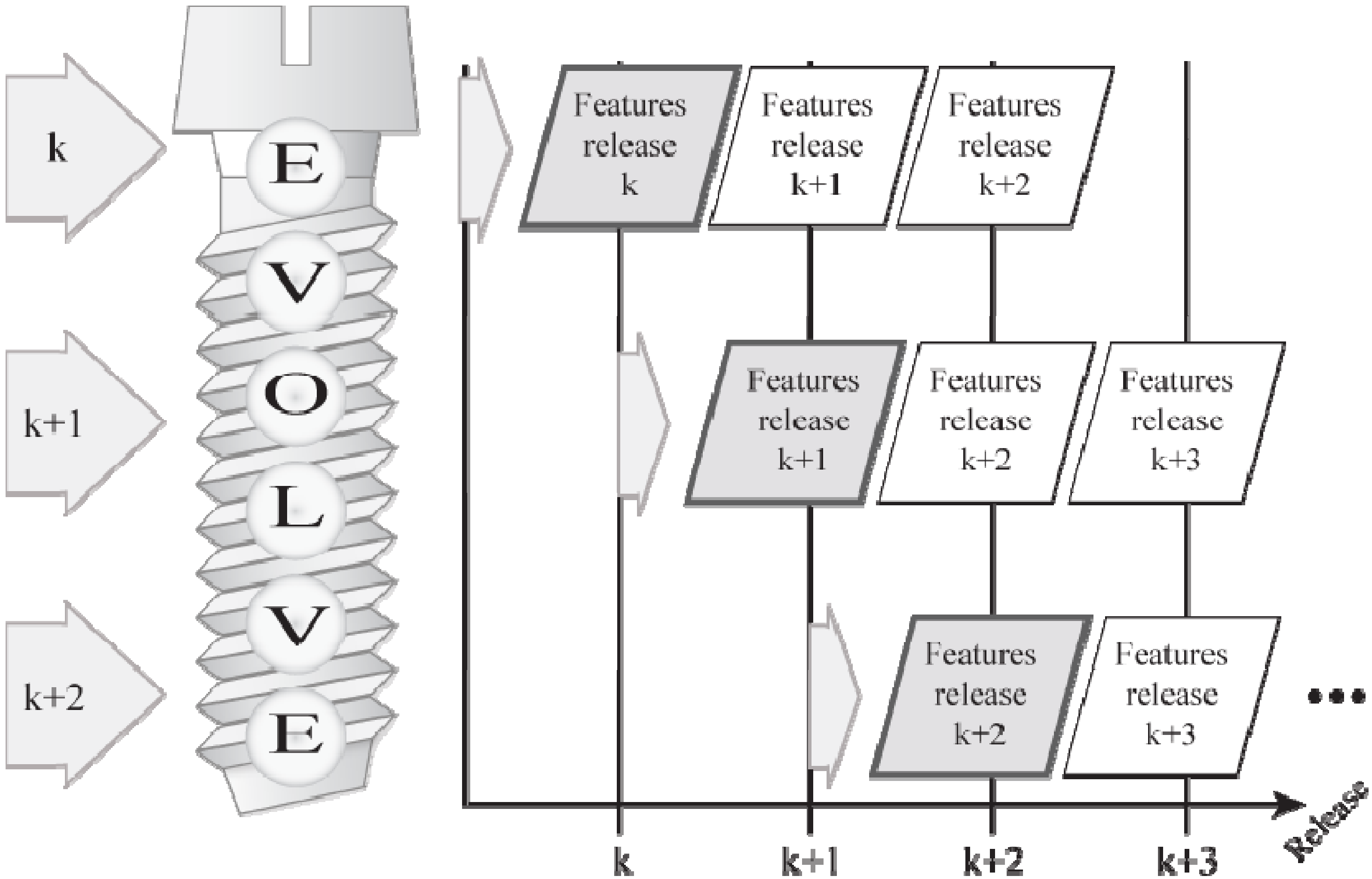
value(n,p) = value_score(n,p)(K – x(n) +1)

EVOLVE (1/2)



Iteration
Input
k
Features
Objectives and Constraints
Priorities
Refined and extended
k+1
Features
Objectives and Constraints
Priorities
Refined and extended
k+2
Features
Objectives and Constraints
Priorities
E
V
O
L
V
E

EVOLVE (2/2)

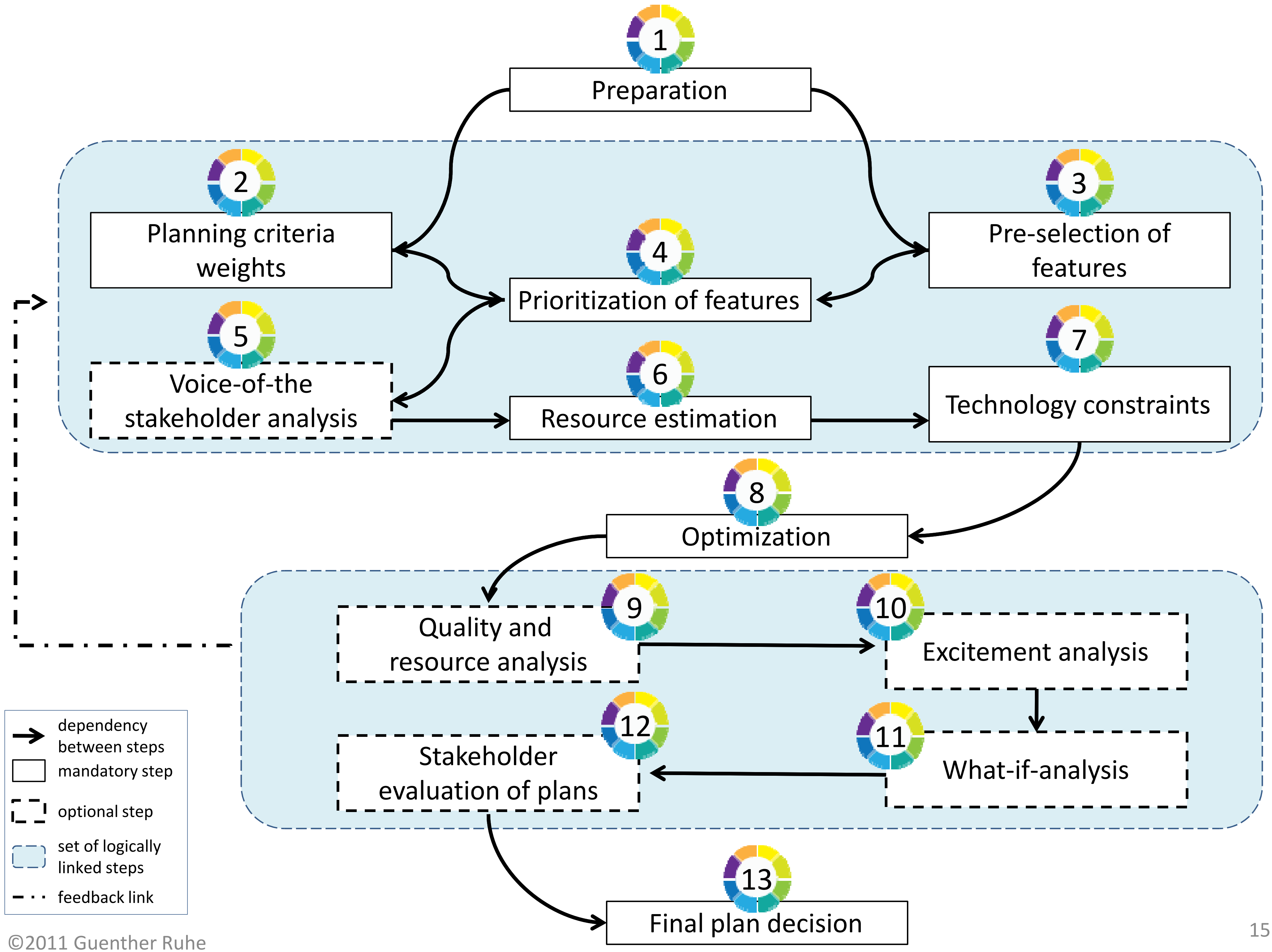


k
k+1
k+2
E
V
O
L
V
E
Features release k
Features release k+1
Features release k+2
Features release k+1
Features release k+2
Features release k+3
Features release k+2
Features release k+3
•••
Release
k
k+1
k+2
k+3

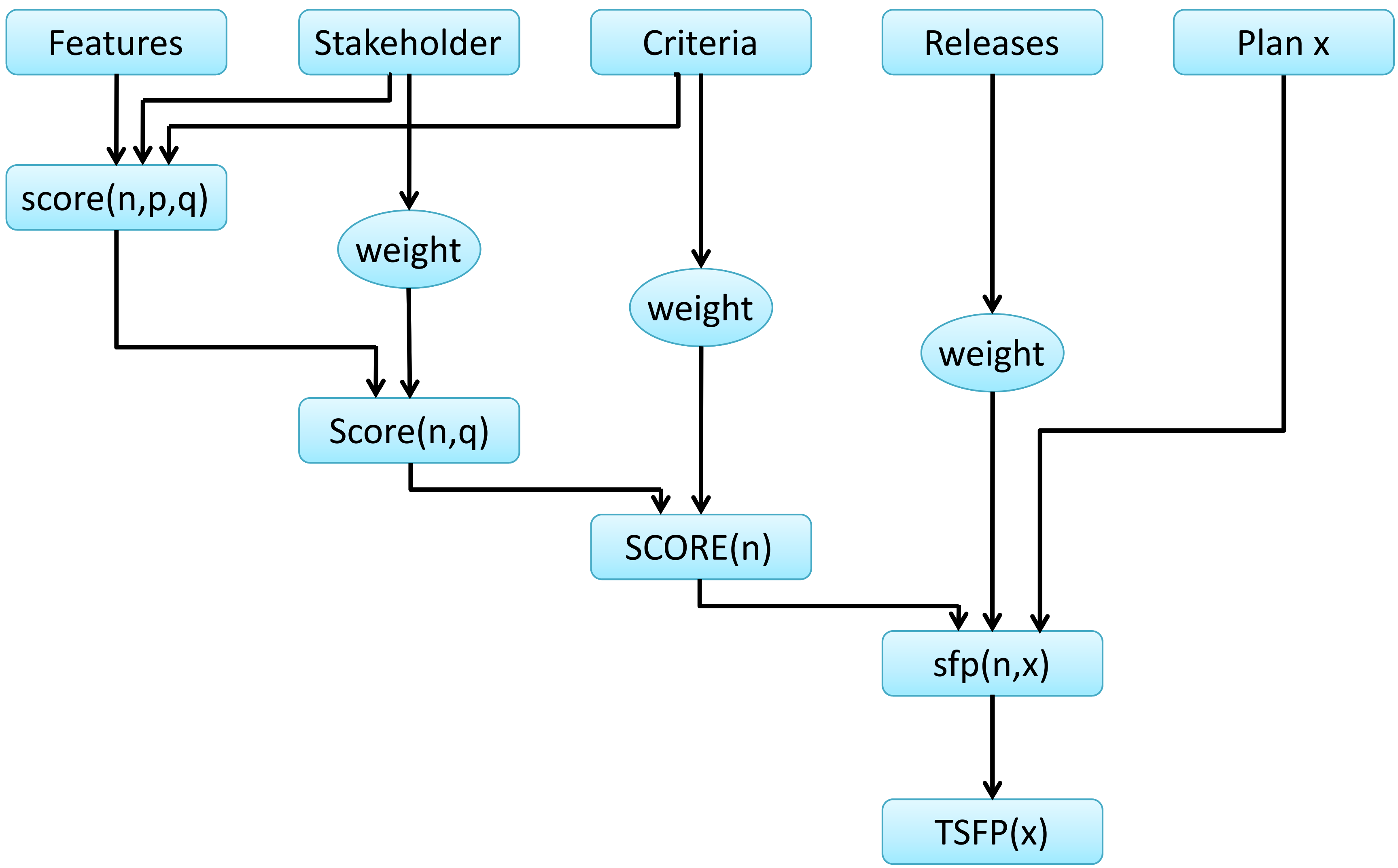
Empirical analysis

- EVOLVE was initially based on genetic search offered by Palisade's RiskOptimizer
- Early industrial feedback (Corel, Siemens)
- Development of our own GA (emphasis on avoiding premature convergence)
- Empirical studies with 200 to 700 requirements comparing the GA with running ILOG's CPLEX
- Better solutions for LP solver in reasonable time
- Known level of optimality
- Development of our own solution method utilizing open source optimization combined with knapsack-type of heuristic for B&B
- New approach more flexible model and with higher level of diversification among top solutions.

Stakeholder-centric release planning – Method EVOLVE II


1
Preparation
2
Planning criteria weights
3
Pre-selection of features
4
Prioritization of features
5
Voice-of-the stakeholder analysis
6
Resource estimation
7
Technology constraints
8
Optimization
9
Quality and resource analysis
10
Excitement analysis
11
What-if-analysis
12
Stakeholder evaluation of plans
13
Final plan decision
dependency between steps
mandatory step
optional step
set of logically linked steps
feedback link

Maximization of stakeholder feature points



Resource constraints

- Resource class 1: A resource type r belongs to class 1 if the feature related consumption of the resource is limited to exactly the release in which the feature if offered. Resources of this class are called local based on its spending mode.

$$\text{Consumption}(k,r,x) = \sum_{n:\ x(n)=k} \text{consumption}(n,r) \leq \text{Capacity}(k,r)$$

Resource constraints

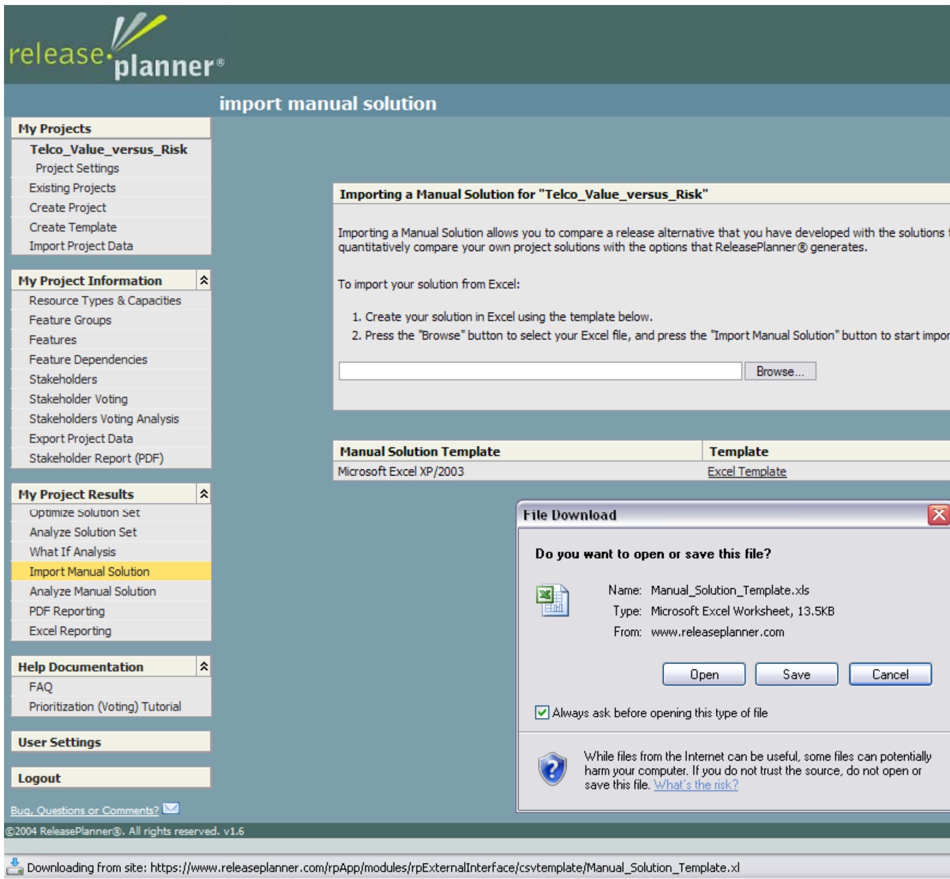
- Resource class 2: A resource type r belongs to class 2 if the feature related consumption of the resource can be distributed across different release periods. Resources of this class are called global based on its spending mode.

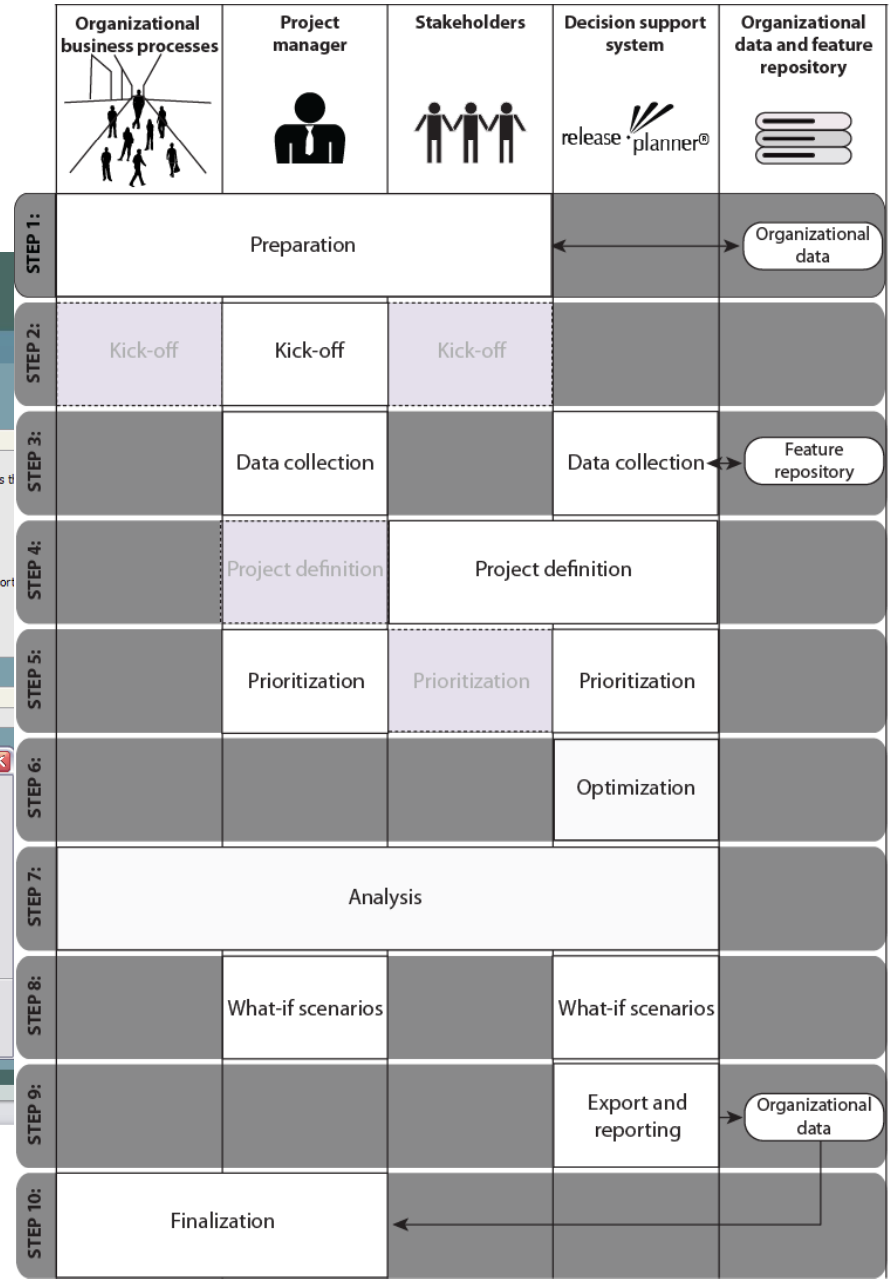
$$\sum_{n=1..N} w_x(n,k,r) \text{ consumption}(n,r) \leq \sum \text{Capacity}(k,r) \text{ for all releases } k = 1\ldots K$$

$$0 \leq w_x(n,k,r) \leq 1 \text{ for all } n,k,r$$

$$\sum_{k=1..K} w_x(n,k,r) = 1 \text{ for all } n,r$$

ReleasePlanner™



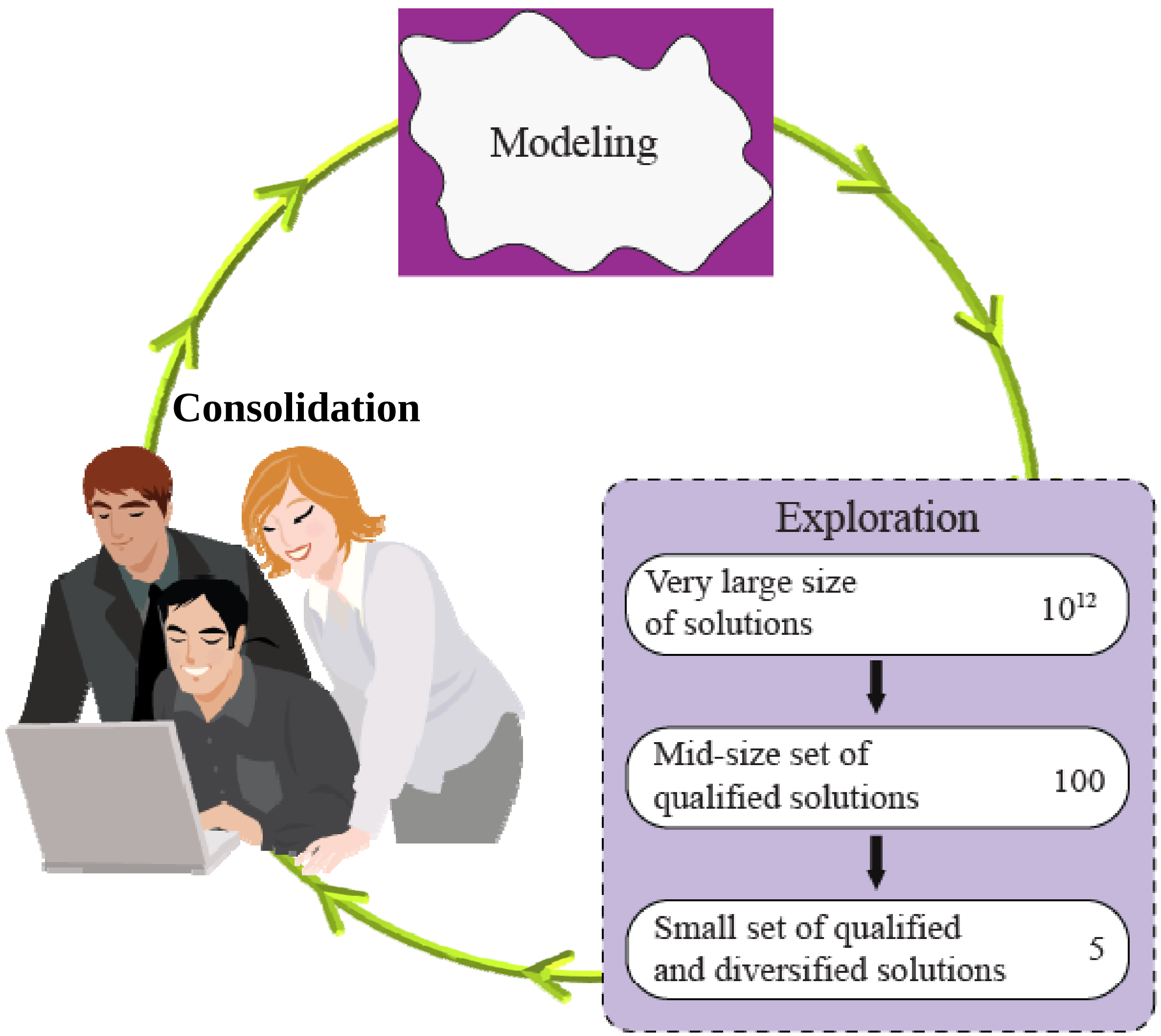


The diversification principle

A single solution

to a cognitive complex problem is less likely to reflect the actual problem when compared to a

portfolio of qualified solutions being structurally diversified



Diversified release plans

INTELLIGENT DECISION SU

PRODUCT LOGOUT CONTACT

Feature Assigned To

1	R14
2	R15
3	Postponed

n set analysis

ucture of plan alternatives

legend]

Feature	Alternative 1	Alternative 2	Alternative 3	Alternative 4	Alternative 5	Manual Solutio
Cost Reduction of Transceiver	1	1	1	1	1	1
16 sector, 12 carrier BTS	3	3	3	1	1	1
Expand Memory on BTS Controller	1	1	1	1	1	1
Next Generation BTS 'In a Shoebox'	3	3	2	3	3	2
Pole Mount Packaging	2	2	3	2	2	3
FCC Out of Band Emissions Regulatory Change	2	1	2	3	2	2
Patching Improvements/Upgrade Enhancementst	3	3	3	2	3	3
CIU and SRM Management Enhancements	1	2	1	3	3	3
SMS Cell Broadcast	1	1	1	1	1	1
Traffic Allocations Enhancements	1	1	1	1	1	1
eBSC CR: CCMC Removal	2	1	2	2	1	1
3 of N Band Class Support	2	1	2	2	1	3
EVRC B Capacity Enhancements	3	3	3	3	3	3
Mobile Recovery Algorithm	2	2	2	3	3	3
Quick Paging Channel Power Offset	3	2	3	3	3	3
Access Optimized IMSI Paging	1	1	1	1	2	3
EBSC REX Testing	2	2	3	1	1	1
CSVS Robustness Enhancements	1	1	1	1	2	1
EBSC Outage Footprint (Flight Recorder)	1	3	1	1	2	2
MFRM Flight Recorder Enhancements	1	3	1	3	2	3

Previous Analysis Step | Next Analysis Step | What If | Export to MS Excel | Restart

https://www.releaseplanner.com/rpApp/?mod=rpRequirement&id=414512 Internet 100%

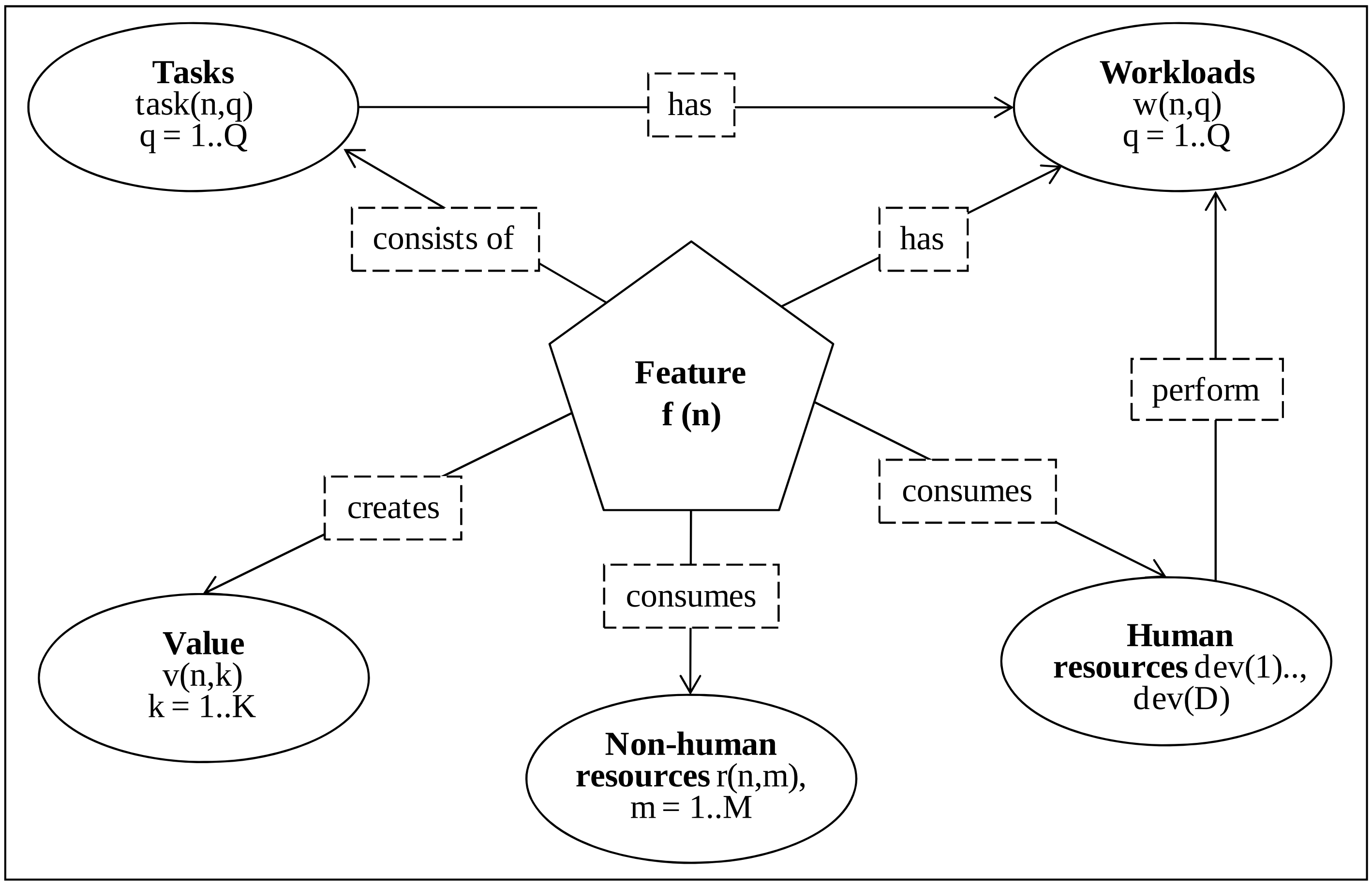
Agenda

1. Decisions in release planning
2. Strategic release planning: Randomized versus deterministic
3. Operational release planning: Deterministic AND randomized
4. When-to-release decisions
5. Ongoing research

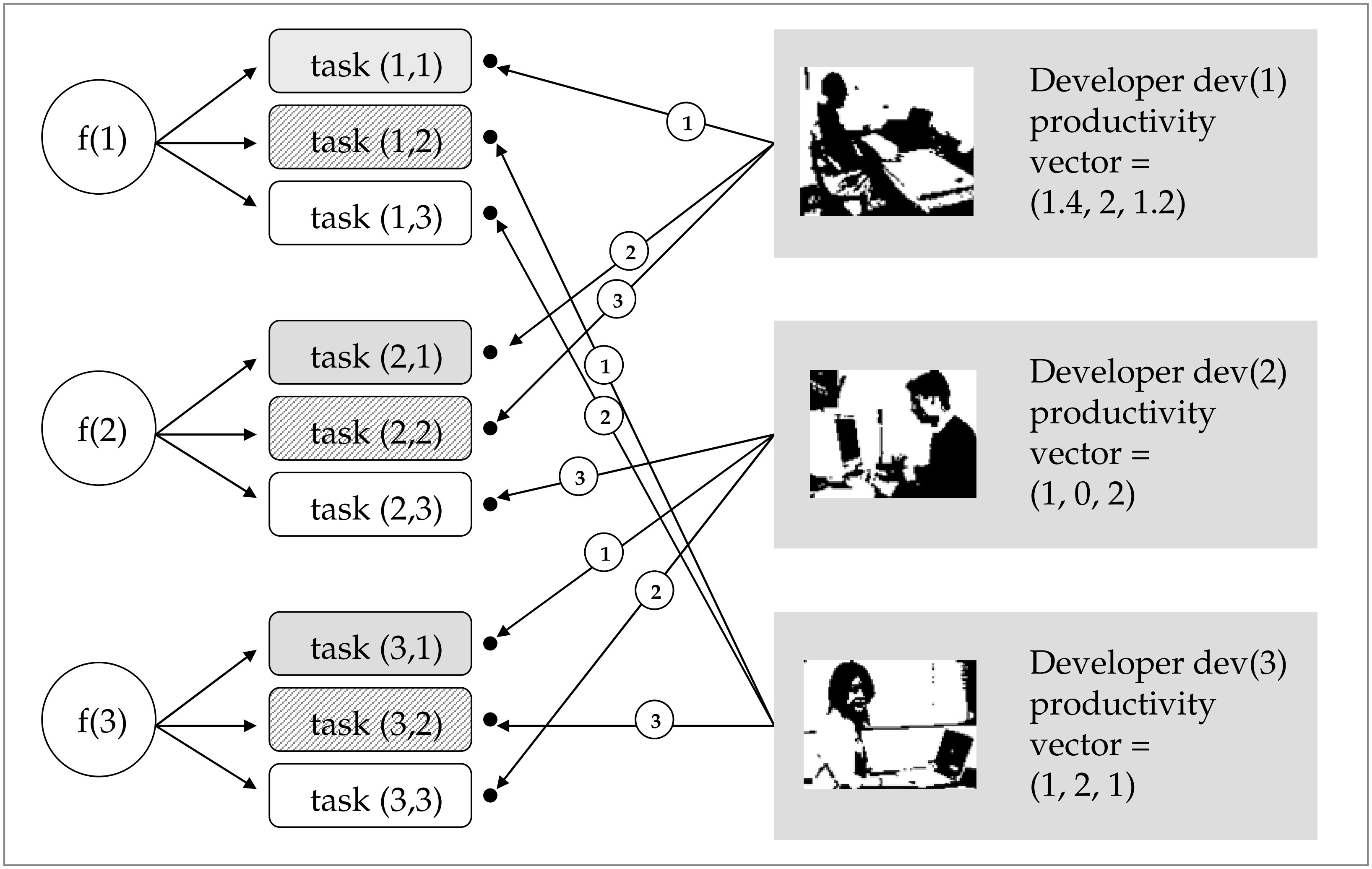
Operational planning: Data related to a feature f(n)

RASORP: A. Ngo-The, G. Ruhe, Optimized Resource Allocation for Software Release Planning, IEEE TSE, Volume 35 (2009), pp 109-123.

P. Kapur, A. Ngo-The, G. Ruhe, A. Smith, Optimized staffing for product releases and its application at Chartwell Technology, JSME Vol.20 (2008), pp 365-386



Assignment of developers to tasks of features



Sample Gantt Chart of operational planning

		Week																							
Feature	Task	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
f(1)	(1,1)	3																							
	(1,2)		3	3	3																				
	(1,3)	5	5		5																				
f(2)	(2,1)	6	6	6	6																				
	(2,2)					1	1																		
	(2,3)						2																		
f(7)	(7,1)			5																					
	(7,2)					3	3	3	3	3															
	(7,3)					5	5	5		5															
f(8)	(8,1)				4																				
	(8,2)							1	1																
	(8,3)					6	6		6																
f(9)	(9,1)					4	4	4																	
	(9,2)									1	1														
	(9,3)							2			2														
f(10)	(10,1)								4	4	4	4													
	(10,2)										3	3	3												
	(10,3)												1												
f(11)	(11,1)							6		6	6	6	6												
	(11,2)								5		5	5	5												
	(11,3)								2	2			2												
f(20)	(20,1)	4	4	4																					
	(20,2)	1	1	1	1																				
	(20,3)	2	2	2	2	2																			
f(3)	(3,1)												4	4											
	(3,2)											1		1	1										
	(3,3)														2										
f(4)	(4,1)													6	6	6									
	(4,2)															1									
	(4,3)													2		2									
f(13)	(13,1)														4	4									
	(13,2)																1	1	1	1	1				
	(13,3)																2	2	2	2	2				
f(14)	(14,1)																4	4							
	(14,2)																	3	3	3	3	3			
	(14,3)																		4	4		4			
f(15)	(15,1)																6	6	6	6					
	(15,2)																		5	5	5	5	5	5	
	(15,3)																				6	6	6	6	
f(17)	(17,1)																				4		4	4	
	(17,2)																					1	1	1	
	(17,3)																					2	2	2	
f(19)	(19,1)											2													
	(19,2)													3	3	3	3								
	(19,3)													5	5	5	5	5							

RASORP modeling (1/3)

- N - number of features under consideration,
- K - number of releases considered for planning,
- Q - number of tasks to be performed for each feature,
- D - number of developers available for assignment to tasks,
- t(k) - due date of release k (k = 1..K), and
- v(n,k) - value obtained by assigning feature n to release k (n = 1..N, k = 1..K)
- x(n,k) - delivery of features f(n) at release k
- u(d,t,n,q) – assignment of developer d at time t to task q of feature f(n)

RASORP modeling (2/3)

- Maximize { $F(x) = \Sigma_{n=1..N\ k=1..K}\ v(n,k) \cdot x(n,k)$ subject to $(u,x) \in UX$} where UX is the set of all feasible combination of staffing and release plans (u,x).
- $\Sigma_{k=1..K}\ x(n,k) \leq 1$ for $n = 1..N$
- $x(n_1,k) = x(n_2,k)$ for all coupled features $C(n_1,n_2)$ for $k = 1..K$
- $\Sigma_{k=1..K}\ (K+1-k)(x(n_1,k) - x(n_2,k)) \geq 0$, for all pairs of features being in precedence relationship $P(n_1,n_2)$
- $\Sigma_{t=t1..t2}\ \Sigma_{n=1..N}\ \Sigma_{q=1..Q}\ u(d,t,n,q) = 0$ for $d = 1..D$, $l = 1..L(d)$, and $twind(d,l) = [t1,t2]$
- $\Sigma_{n=1..N}\ \Sigma_{q=1..Q}\ u(d,t,n,q) \leq 1$ for $d = 1..D$ and $t = 1..t(K)$

RASORP modeling (3/3)

- $\Sigma_{t=1..t(K)}$ u(d,t,n,q) ≤ t(K)*z(d,n,q) for d= 1..D, n = 1..N and q = 1..Q
- $\Sigma_{d=1..D}$ z(d,n,q) ≤ 1 for n = 1..N and q = 1..Q
- $\Sigma_{d=1..D}$ $\Sigma_{t=1..t(k)}$ u(d,t,n,q)*prod(d,q) ≥ w(n,q)*$\Sigma_{k1=1..k}$ x(n,k_1) for k = 1..K, n = 1..N and q = 1..Q
- $\Sigma_{d=1..D}$ $\Sigma_{t1=1..t}$ u(d,t1,n,q) ≥ $\Sigma_{d=1..D}$ $\Sigma_{t1=1..t}$ u(d,t1,n,q+e) for t = 1..t(K), n = 1..N, q = 1..Q-1,
  w(n,q), w(n,q+e) > 0 and
  w(n,q*) = 0 for all q* = q+1..q+e-1
- $\Sigma_{d=1..D}$ $\Sigma_{t1=t..t(K)}$ u(d,t1,n,q+e) ≥ $\Sigma_{d=1..D}$ $\Sigma_{t1=t..t(K)}$ u(d,t1,n,q) for t = 1..t(K), n = 1..N, q = 1..Q-1, e = 1..Q – q,
  w(n,q), w(n,q+e) > 0 and
  w(n,q*) = 0 for all q* = q+1..q+e-1

RASORP algorithm - Phase 1 (packaging)

- Step 1.1
  Consider a simplified problem formulation RASORP* by ignoring the precedence constraints between the tasks implementing the features (just looking at t = t(k)'s).
- Step 1.2
  Apply branch and bound techniques in combination with linear programming (solving the relaxed problem without integrality constraints) to generate upper bounds and using a greedy heuristic to solve the sub-problem at each node of the branching tree.
- Step 1.3
  Obtain an optimized solution x1 which is taken as an input for Phase 2 to define a reduced search space.

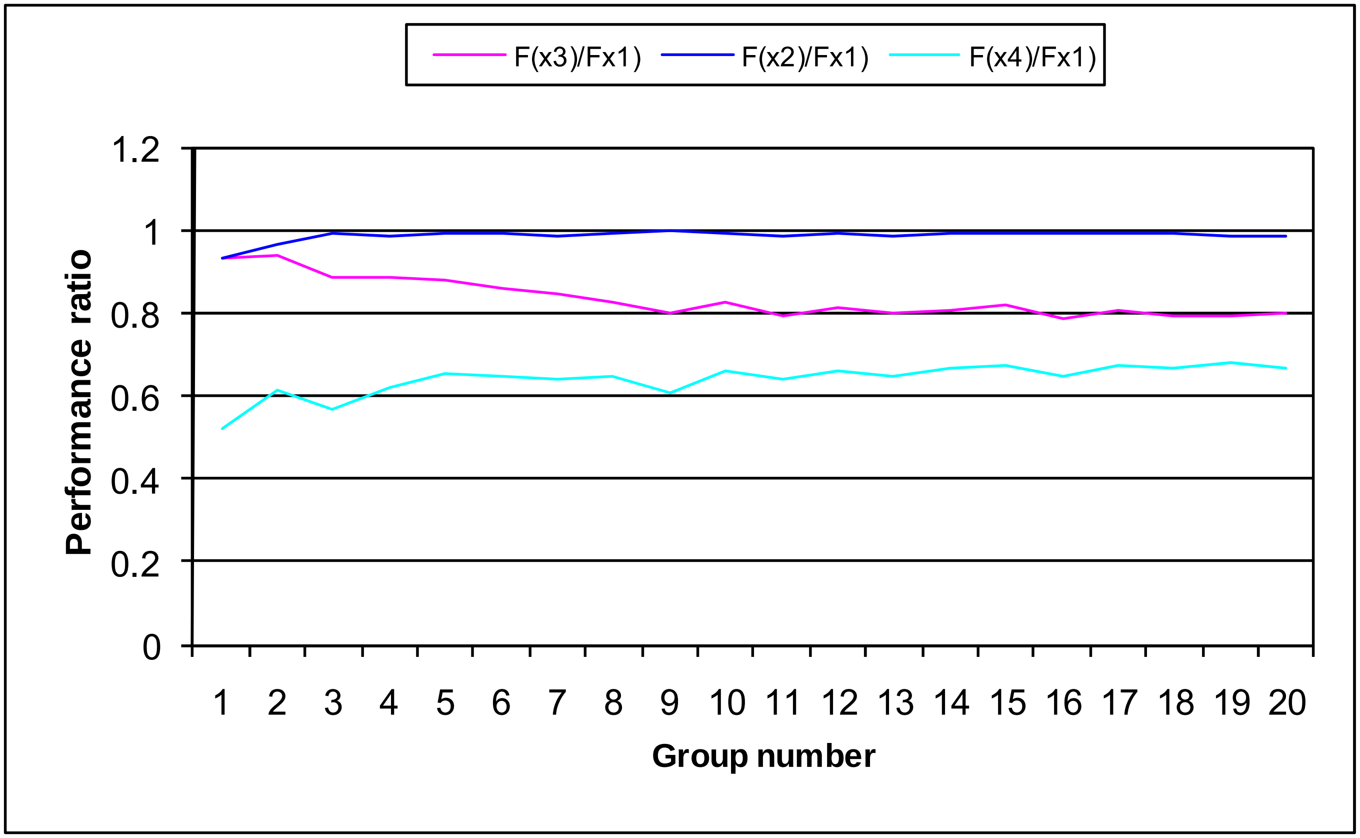
RASORP algorithm Phase 2 (scheduling)

- Step 2.1: Consider the complete problem RASORP.
- Step 2.2: Apply genetic algorithms to a reduced search space of permutations called ∏* (focused search) defined by the solution x1 from Phase 1.
  - Population size = 100,
  - Maximal number of generations = 500,
  - Probability of mutation = 1%,
  - Termination: If there is no improvement after 100 consecutive generations or the maximum number of generations is reached,
  - Percentage of new random solutions in each new generation= 10%,
  - Number of generations indicating that the population is stuck at a local optimum = 50, and
  - Proportion of new individuals when the population is stuck at a local optimum = 80%.
- Step 2.3: The resulting solution x2 has a degree of optimality of at least F(x2)/ F(x1).

Empirical analysis: Definition of groups

Group	Range N	Average K	Average M	Average Q	Average D	Average lPl	Average HR	Average NHR
1	5-14	2.667	1.852	3.333	2.111	0.111	0.689	0.681
2	15-24	2.296	1.704	3.333	2.333	0.481	0.689	0.696
3	25-34	3.037	1.778	3.333	3.111	0.667	0.693	0.707
4	35-44	2.148	1.926	3.333	3.370	1.852	0.715	0.685
5	45-54	2.259	1.741	3.333	3.741	1.889	0.726	0.704
6	55-64	2.630	1.778	3.333	4.741	3.000	0.711	0.726
7	65-74	2.333	1.889	3.333	5.407	2.889	0.689	0.722
8	75-84	2.444	1.741	3.333	5.333	3.704	0.711	0.715
9	85-94	2.815	1.926	3.370	7.074	3.444	0.674	0.704
10	95-104	2.370	1.704	3.333	7.037	5.074	0.693	0.730
11	105-114	2.667	1.926	3.667	8.111	6.185	0.711	0.719
12	115-124	2.370	1.704	3.667	8.926	6.741	0.711	0.719
13	125-134	2.111	1.889	3.667	10.556	7.111	0.685	0.700
14	135-144	2.111	1.926	3.667	10.815	7.259	0.674	0.715
15	145-154	2.148	2.037	3.630	13.519	6.037	0.726	0.719
16	155-164	2.481	1.963	3.630	13.593	7.852	0.707	0.674
17	165-174	2.148	2.000	3.630	15.000	9.444	0.700	0.704
18	175-184	2.074	1.852	3.630	13.963	10.593	0.681	0.693
19	185-194	1.963	1.778	3.519	14.111	9.667	0.696	0.719
20	195-204	1.963	1.926	3.630	17.074	8.963	0.719	0.719

Comparison between FS, UFS and greedy search



- x1 = optimized plan at the end of Phase 1
- x2 = plan received from application of focused search FS
- x3 = plan received from application of unfocused search UFS
- x4 = plan received from application of greedy search

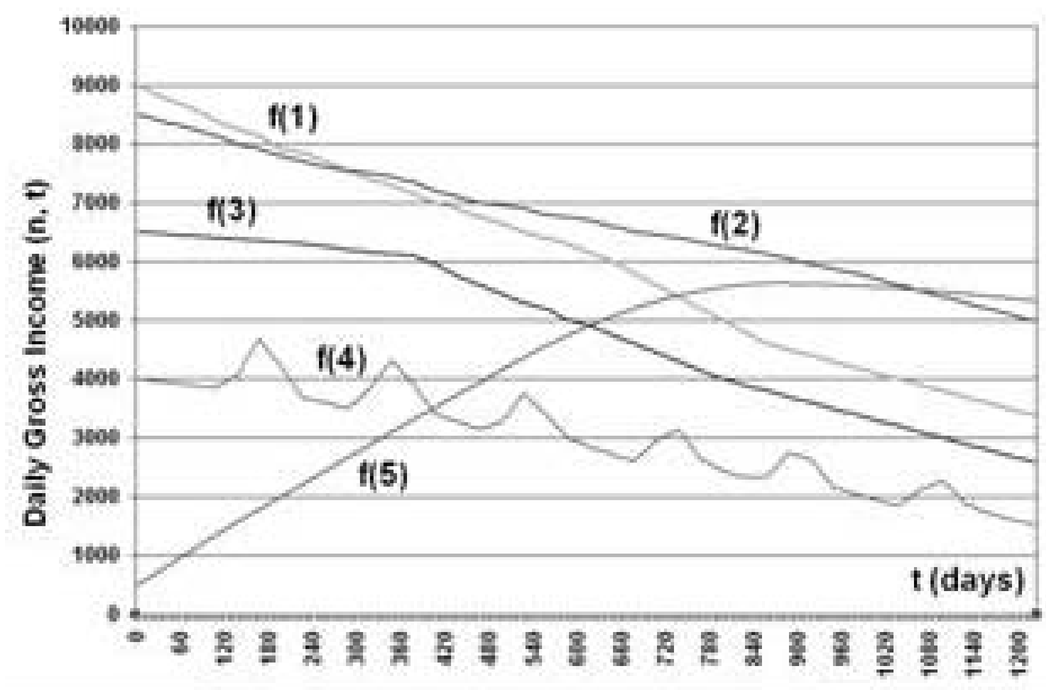
Agenda

1. Decisions in release planning
2. Strategic release planning: Randomized versus deterministic
3. Operational release planning: Deterministic AND randomized
4. When-to-release decisions
5. Ongoing research

When-to-release decisions for features with time-dependent value functions

J. McElroy, G. Ruhe: When-to-release Decisions for Features with Time-dependent Value Functions, Requirements Engineering Journal, Requirements Engineering, Vol. 15 (2010), pp. 337-358

- Value functions are are continuous functions of time.



$$TNV(n,t) = \int_t^T DailyGrossIncome(n,t)\, dt$$

When-to-release decisions for features with time-dependent value functions

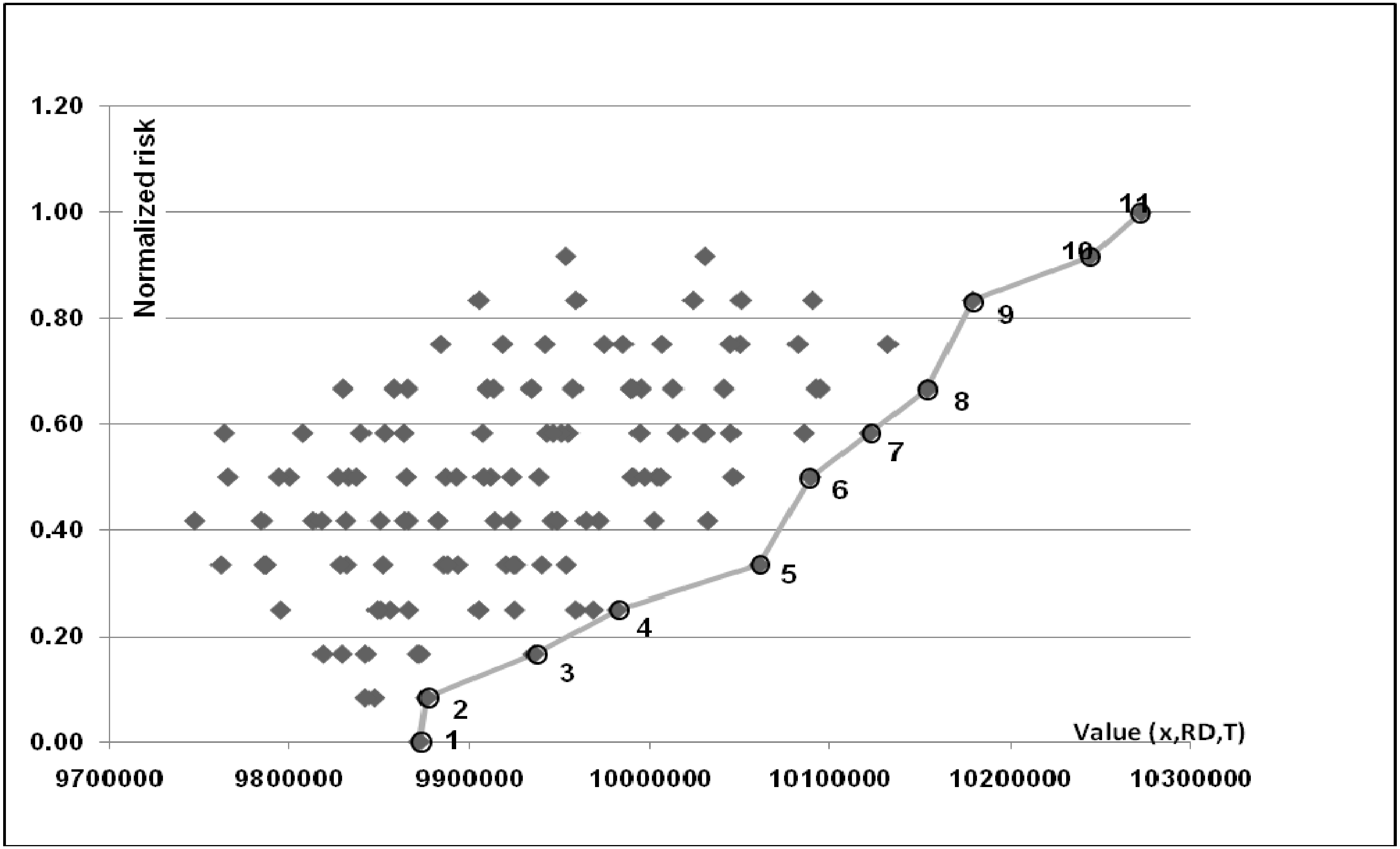
- Actual release dates are no longer fixed but can be varied in some pre-defined interval.

  Consumption(k,r,x) = $\sum_{n:x(n)=k}$ consumption(n,r) $\leq$ Capacity(k,r,t*(k)) for r = 1…R and k = 1…K

  t*(k) being from the interval [rd1(k), rd2(k)]
- Value(x, RD,T) = $\Sigma_{k=1\ldots K}\ \Sigma_{n:\ x(n)=k}$ TNV(n,rd(k))
- Risk(x, RD) = $\sum_{k\,=\,1..K} \alpha(k)\ [rd2(k) - rd(k)]^{\beta(k)}$
- Calculation of trade-off solutions balancing the risk of early release with the potential additional value.

  Trade-off {[Value(x, RD,T) , Risk(x, RD))] according to
  x $\in$ X(RD), RD = (rd(1)…rd(K)) with
  rd(k) $\in$ [rd1(k), rd2(k)] for all k = 1…K

Risk-value trade-off solutions



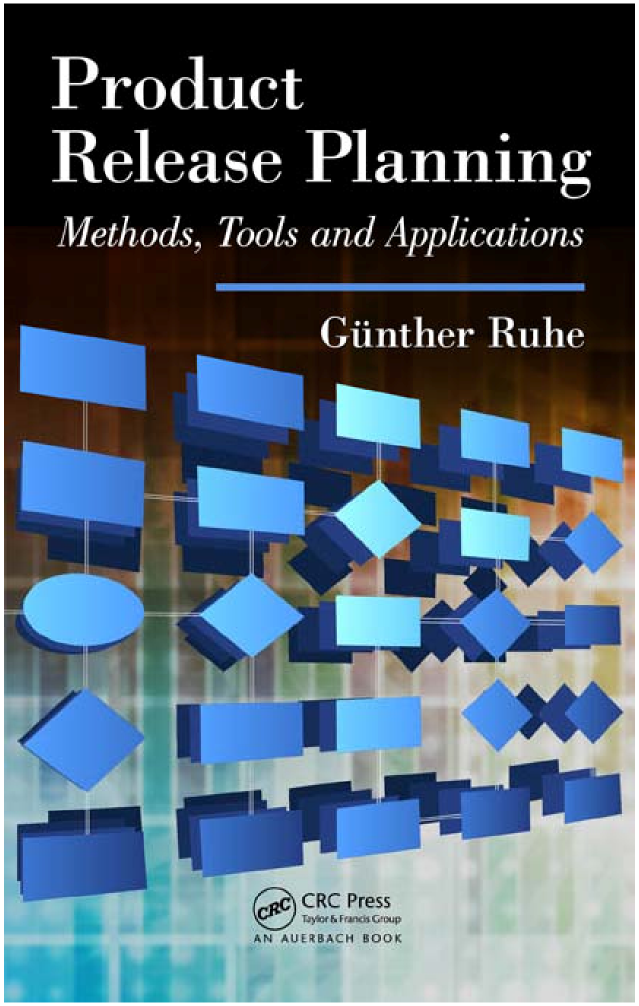
Agenda

1. Decisions in release planning
2. Strategic release planning: Randomized versus deterministic
3. Operational release planning: Deterministic AND randomized
4. When-to-release decisions
5. Ongoing research

Topics of ongoing and future research

- Release planning with logical constraints
- Planning and mining for software releases
- Planning for multiple products
- Release decisions in consideration of the impact of uncertainty
- Release decisions in consideration of functional and non-functional requirements

A bit of self advertisement



ISBN: 9780849326202
ISBN 10: 0849326206

CRC Press
Taylor & Francis Group

Contact:

Guenther Ruhe
ruhe@ucalgary.ca

Empirical Software Engineering International Week

September 19-23, 2011 - Banff, Alberta, Canada

eseiw2011

Empirical Software Engineering International Week (ESEIW) is comprised of six conferences/events:

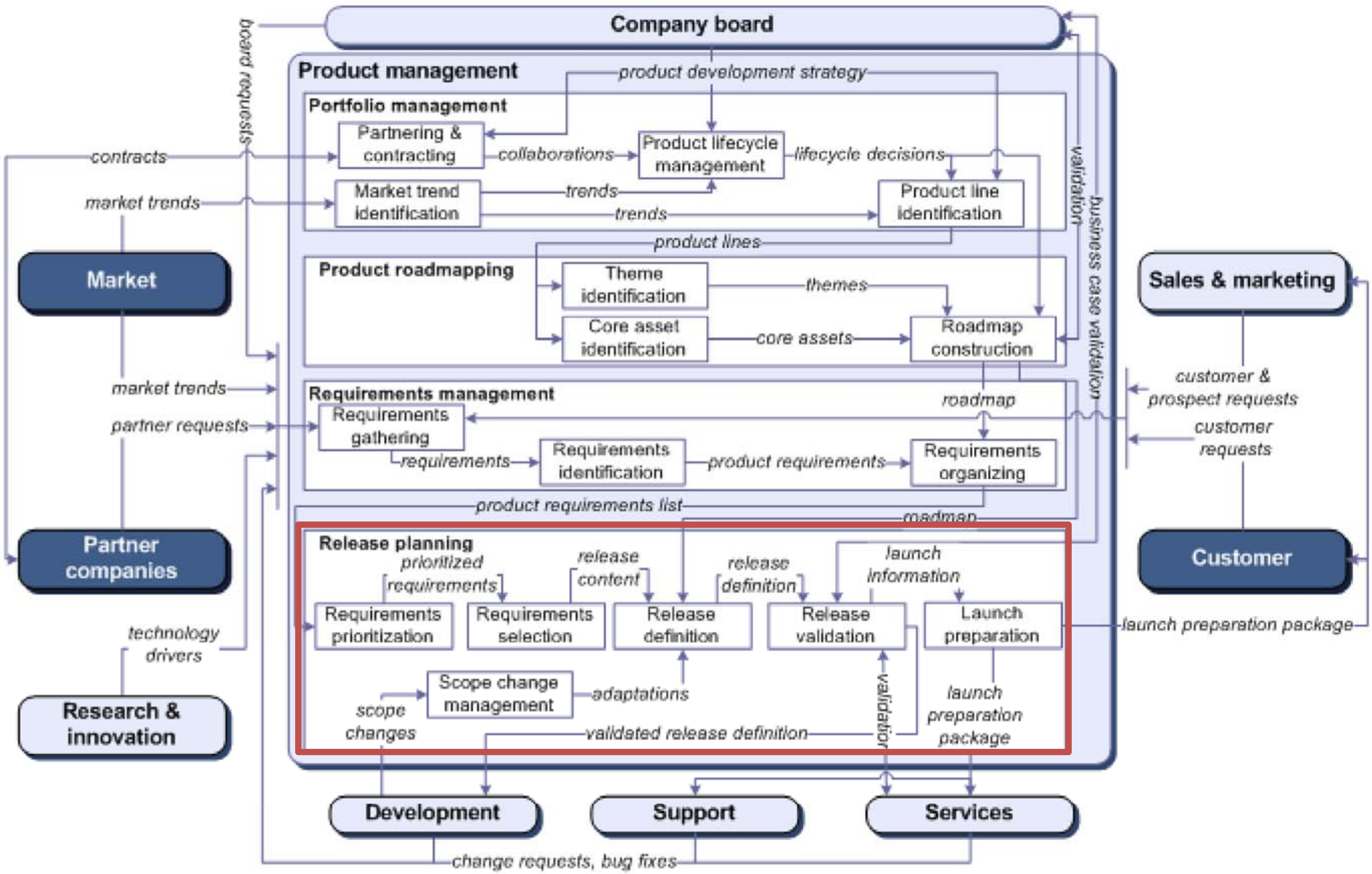
Abbrev./Link	Program	Full Name	Location(s)
ESEM	Link	International Symposium on Empirical Software Engineering and Measurement	
ISERN	Link	International Software Engineering Research Network	
IDoESE	Link	International Doctoral Symposium on Empirical Software Engineering	
IASESE	Link	International Advanced School on Empirical Software Engineering	
PROMISE	Link	International Conference on Predictive Models in Software Engineering	
RESER	Link	International Workshop on Replication in Empirical Software Engineering Research	
MetriSec	Link	International Workshop on Security Meaurements and Metrics	

The schedule for ESEIW is as follows.

September 2011

Mon 19	Tue 20	Wed 21	Thu 22	Fri 23
ISERN	ISERN	IDoESE	ESEM	ESEM
		IASESE		
	PROMISE	PROMISE		
		RESER		
		MetriSec		

The bigger picture



Source: http://www.softwareproductmanagement.org/

Discussion

- It is more important to solve the right problem instead of solving a problem right
- Modeling is more influential than solving
- “Traditional” optimization has advantages, too
- Huge gap in transferring research results into industry
- More evidence for usefulness is needed