9th International Advanced School of
Empirical Software Engineering
September 21, 2011 - Banff, Alberta, Canada

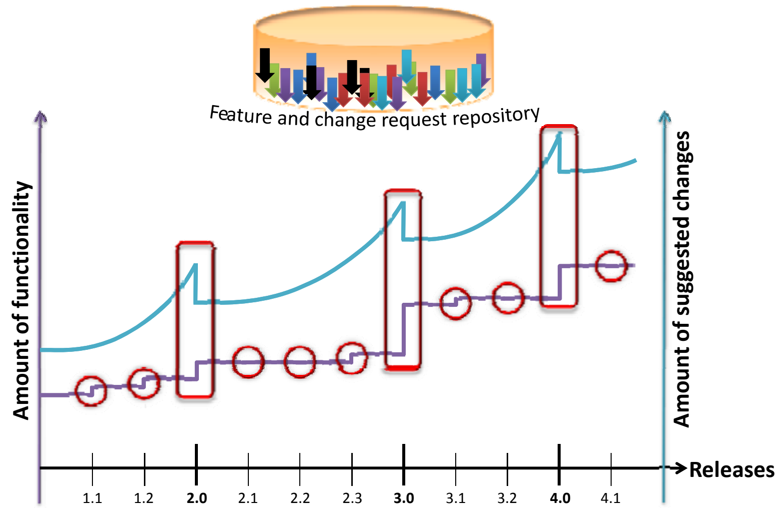
Software release and version Planning – A decision-centric approach

Guenther Ruhe
University of Calgary
Canada

Agenda

1. Decisions in release and version planning
2. The process of decision-making
3. Strategic release planning: Randomized versus deterministic
4. Operational release planning: Deterministic AND randomized
5. When-to-release decisions
6. Summary and outlook

Release and version planning – What it is?



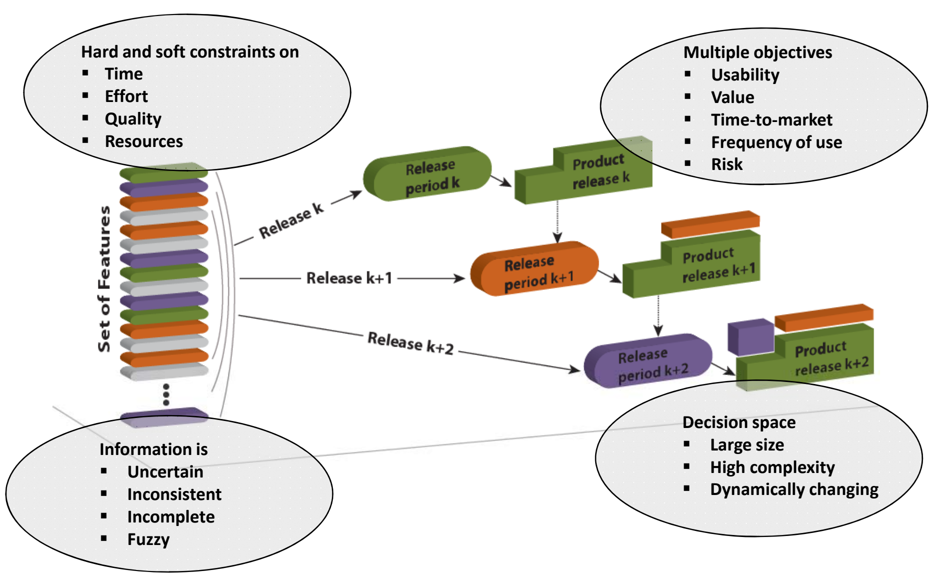
Release and versioning decisions

- Which features should 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?
- When to create a customized version of a product release?

Key components of a decision

- Decision (independent) variables
  - determine which decision is to be made
  - are used to formulate constraints that have to be (or that should be) fulfilled by the decisions
  - are used to formulate objective(s) of decision-making.
- Result (outcome) variables
  - defines the results of decision-making
  - quantitative or non-quantitative relationship to decision variables.
- Uncontrolled variables
  - environmental factors that influence the decisions and their results, but are not/hard to control.

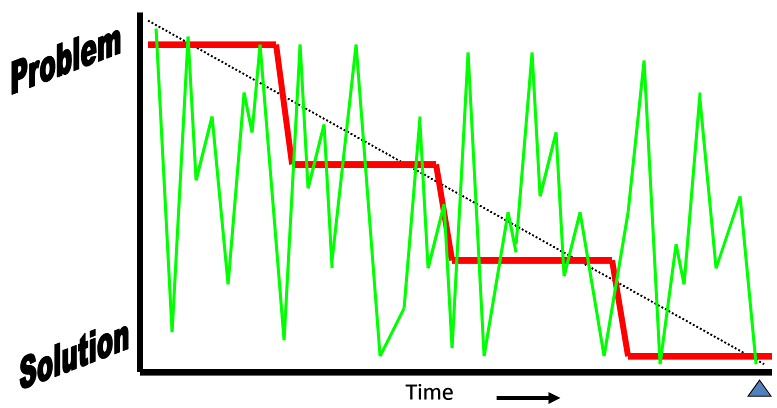
Release planning and versioning - Why difficult?



Agenda

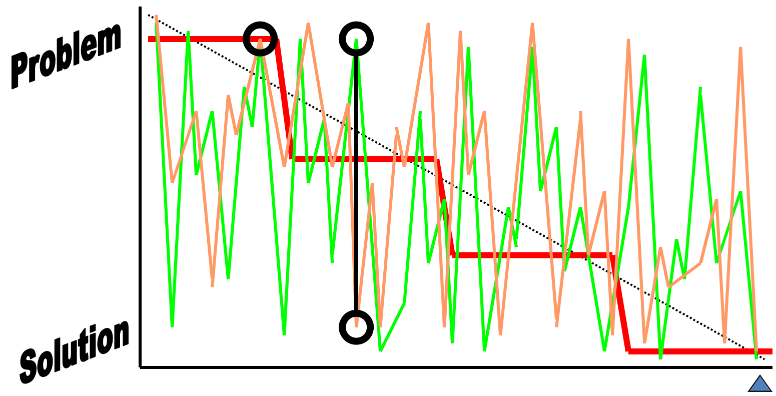
1. Decisions in release and version planning
2. The process of decision-making
3. Strategic release planning: Randomized versus deterministic
4. Operational release planning: Deterministic AND randomized
5. When-to-release decisions
6. Summary and outlook

An experiment about human problem solving (1/2)



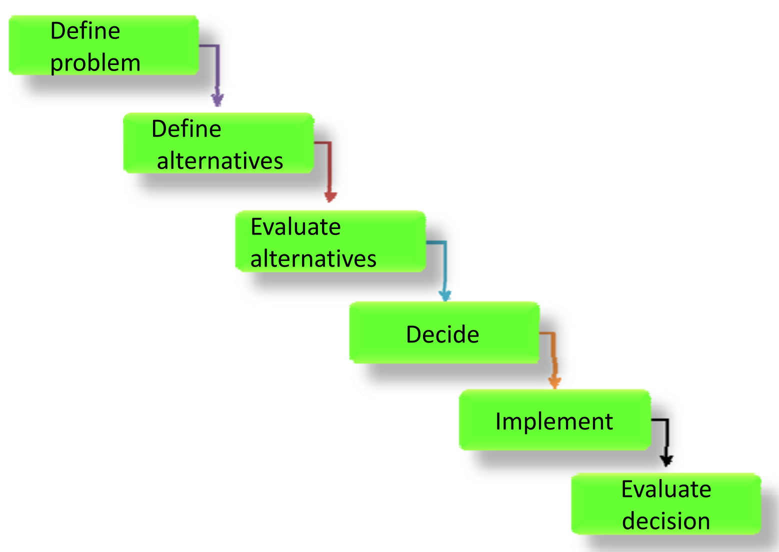
[1)] Jeff Conklin CogNexus Institute

An Experiment about humans problem solving (2/2)

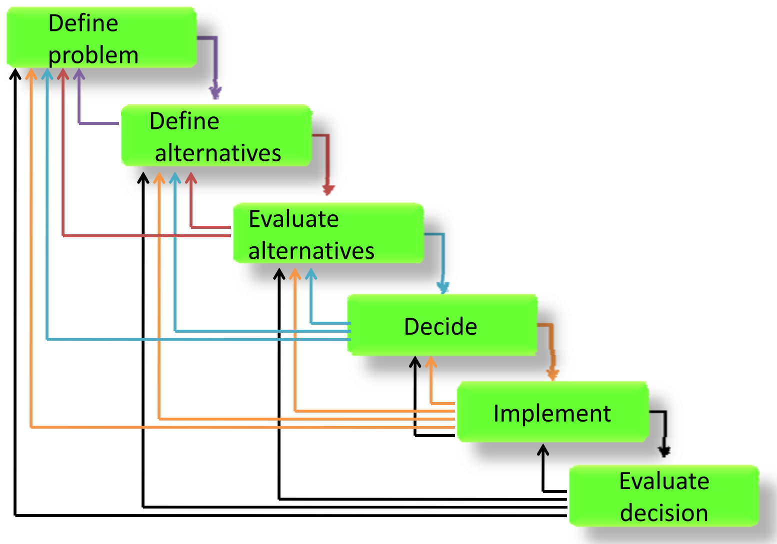


[1] Jeff Conklin CogNexus Institute

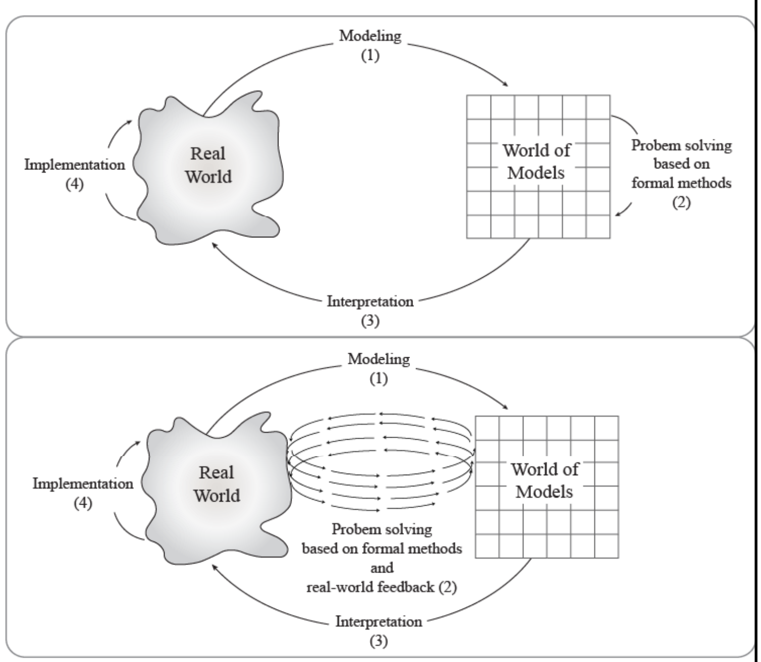
Decision process – Ideal view



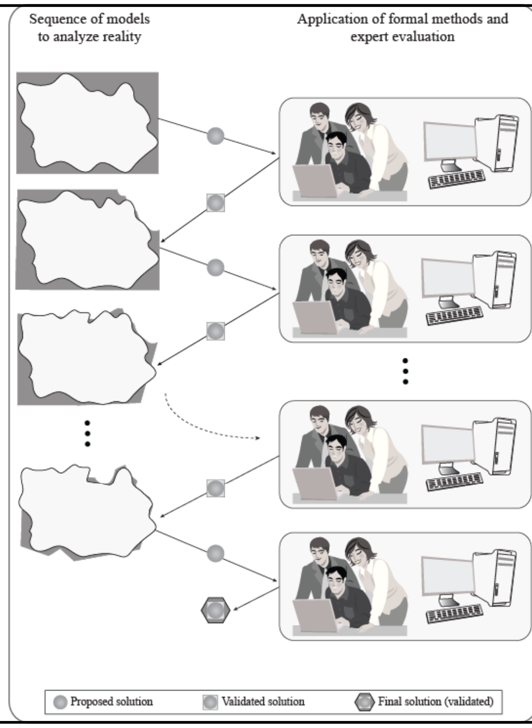
Decision process – Realistic view



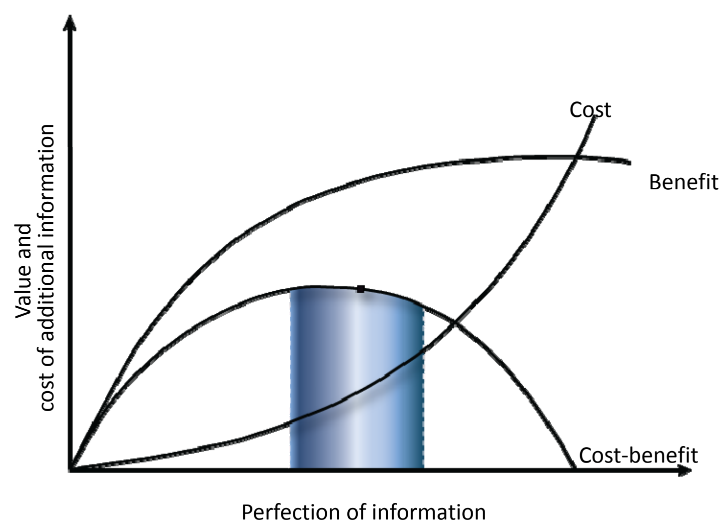
The principal approach of problem solving without (top) resp. with feedback (bottom)



Evolutionary problem solving process



Cost-benefit analysis of investment into decision support



ROI the better the more often investments are used

(Harrison 1987)

Agenda

1. Decisions in release and version planning
2. The process of decision-making
3. Strategic release planning: Randomized versus deterministic
4. Operational release planning: Deterministic AND randomized
5. When-to-release decisions
6. Summary and outlook

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.

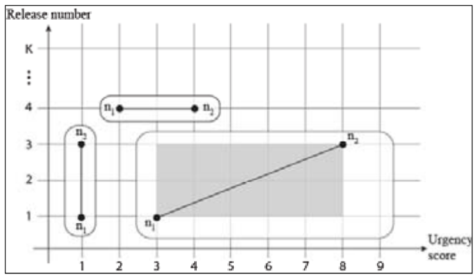
What constitutes a release plan?

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

- 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 monotonicity property between all pairs of features

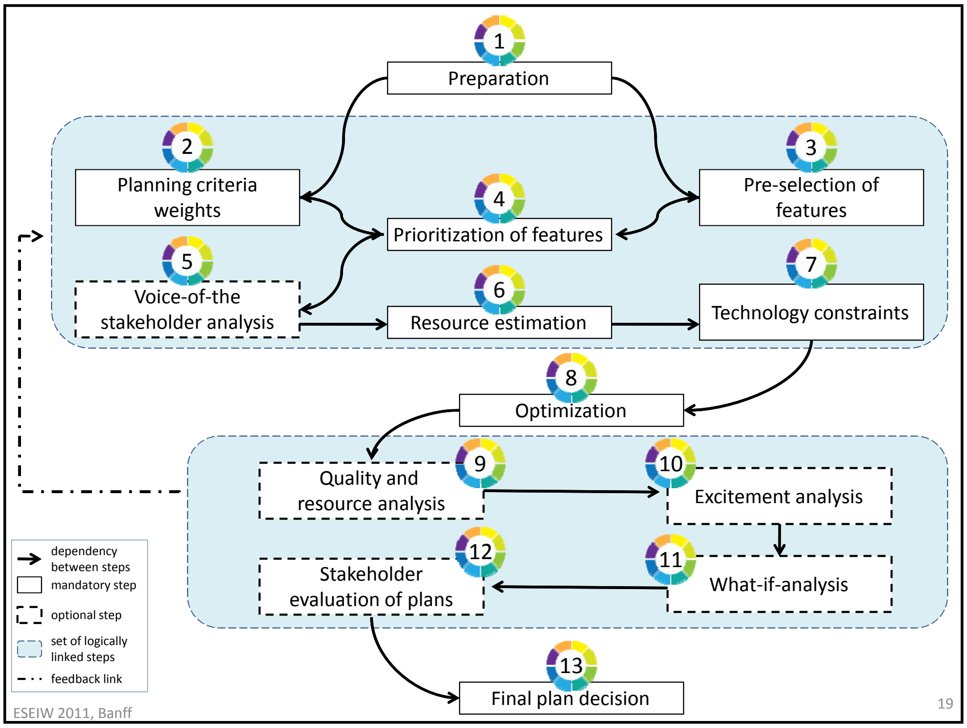


F2(x) is a benefit function based on feature scores of the stakeholders and the actual assignment of the feature according to the plan under consideration.

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

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 based upon a more flexible model and with higher level of diversification among top solutions.



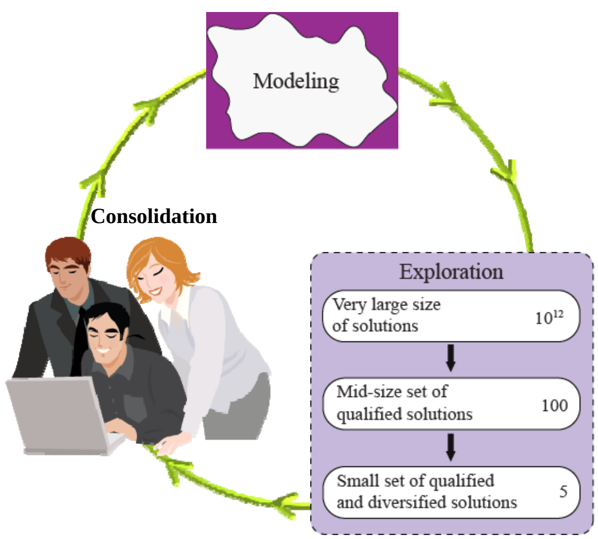
ESEIW 2011, Banff 19

The diversification principle

A single solution

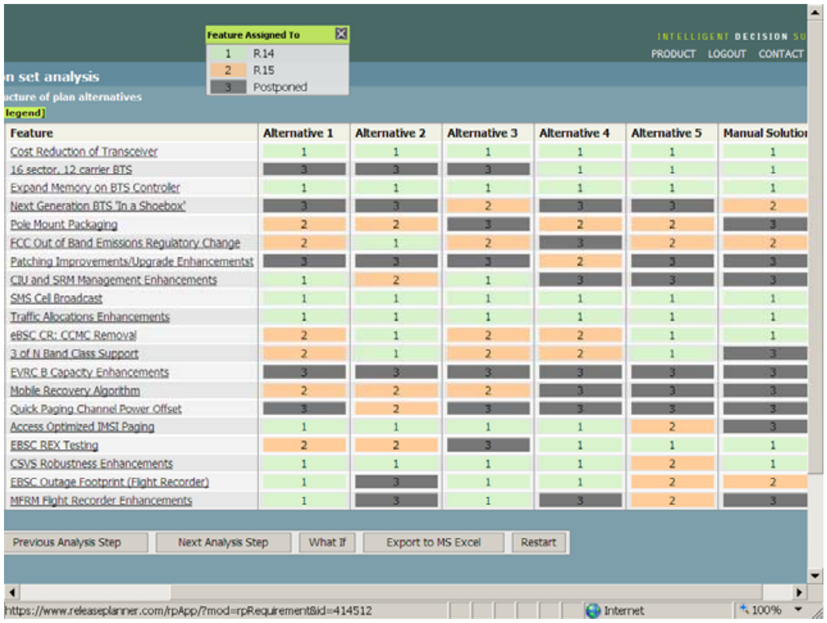
to a cognitive complex problem is less likely to reflect the actual problem when compared to a

portfolio of qualified solutions being structurally diversified



ESEIW 2011, Banff 20

Diversified release plans



Feature	Alternative 1	Alternative 2	Alternative 3	Alternative 4	Alternative 5	Manual Solutior
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

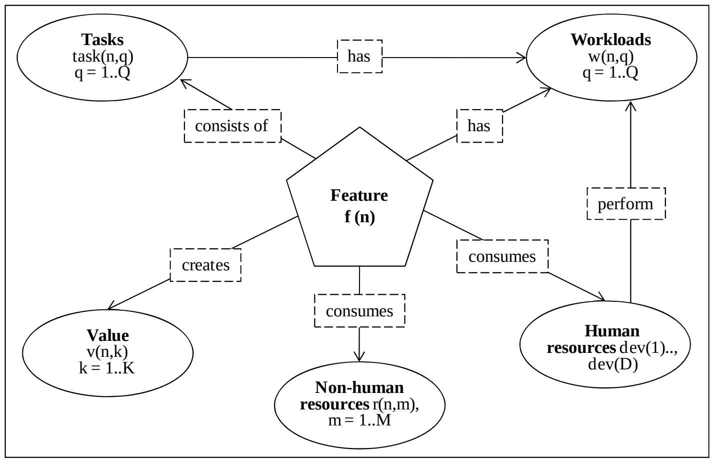
Agenda

1. Decisions in release and version planning
2. The process of decision-making
3. Strategic release planning: Randomized versus deterministic
4. Operational release planning: Deterministic AND randomized
5. When-to-release decisions
6. Summary and outlook

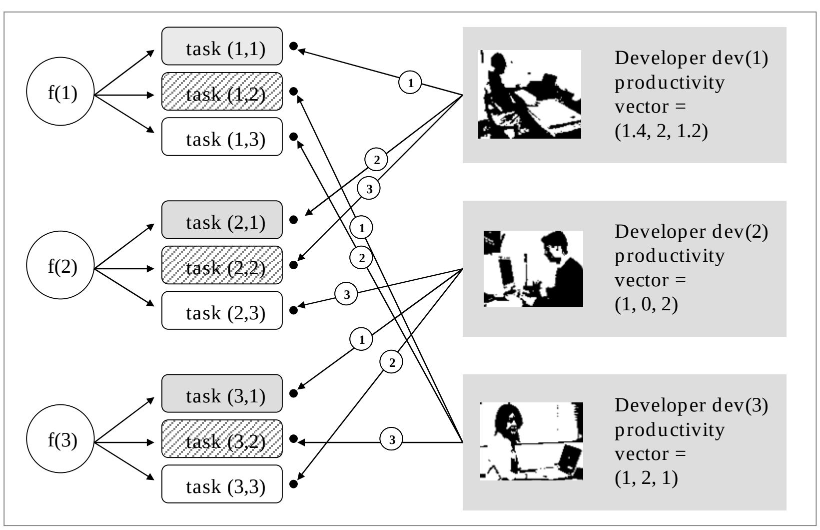
Operational planning: Data related to a feature f(n)

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

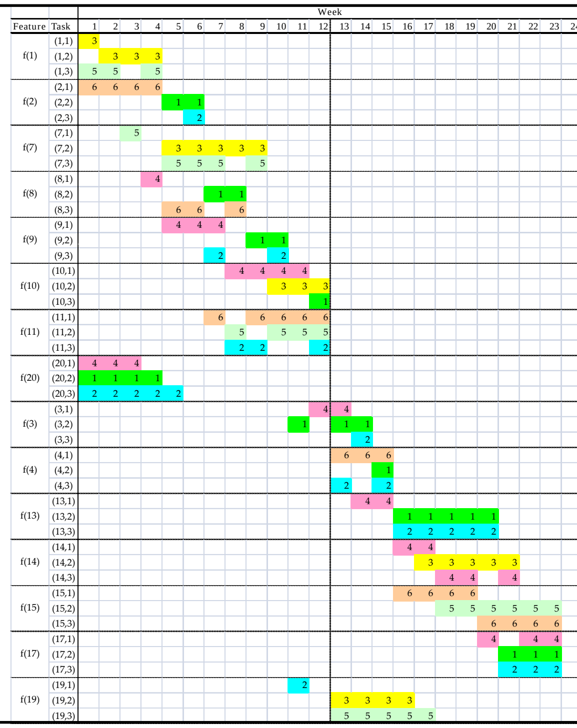
P. Kapur, A. Ngo-The, G. Ruhe, A. Smith, Optimized staffing for product releases and its application at Chartwell Technology, JSME Vol.20 (2008), pp 365-386



Assignment of developers to tasks of features



Sample Gantt Chart of operational planning



"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)

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.

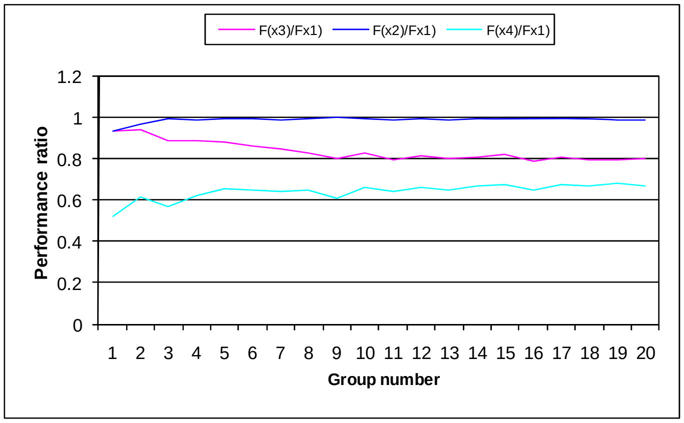
RASORP algorithm Phase 2 (scheduling)

- 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

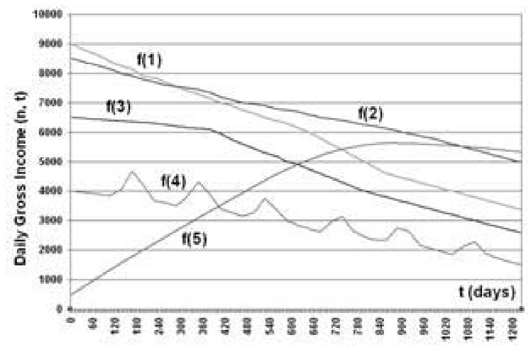
Agenda

1. Decisions in release and version planning
2. The process of decision-making
3. Strategic release planning: Randomized versus deterministic
4. Operational release planning: Deterministic AND randomized
5. When-to-release decisions
6. Summary and outlook

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

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

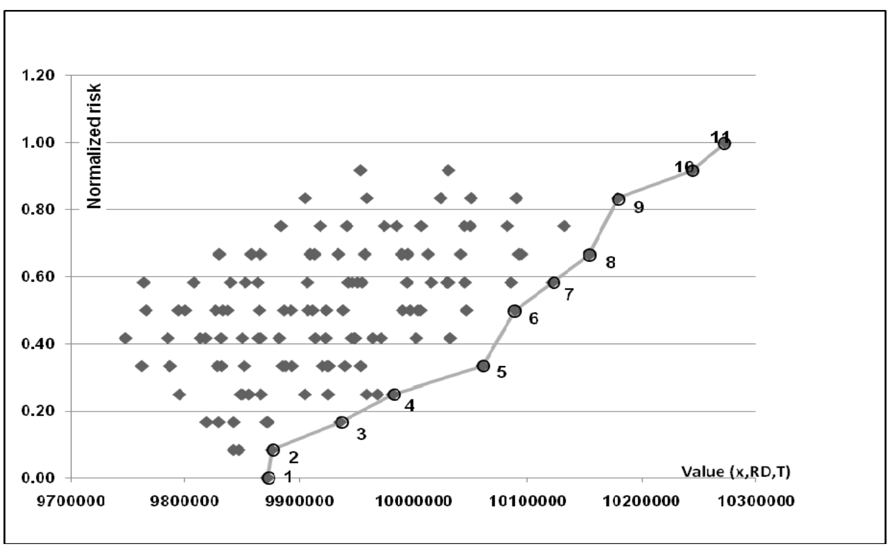
- Actual release dates are no longer fixed but can be varied in some pre-defined interval.

  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) = $\Sigma_{k=1...K} \Sigma_{n: x(n) = k}$ 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

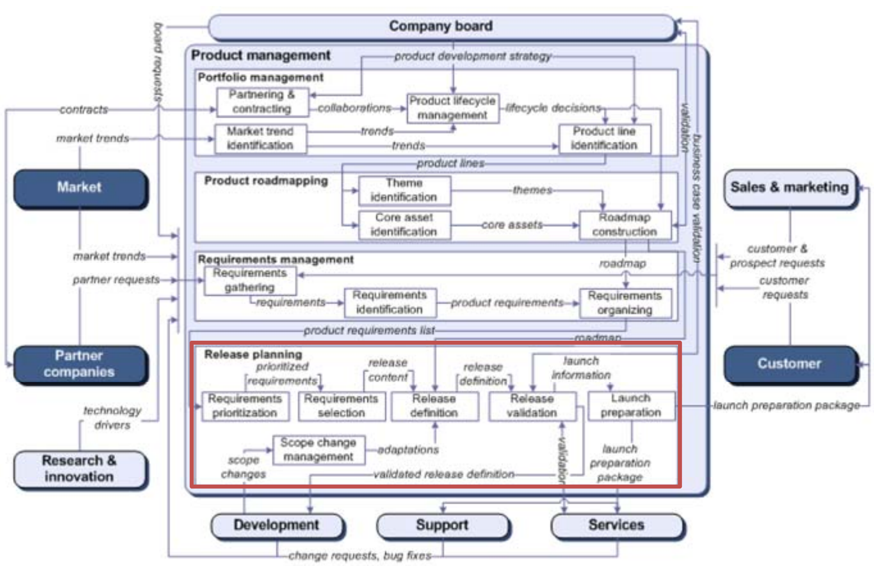
Risk-value trade-off solutions



Agenda

1. Decisions in release and version planning
2. The process of decision-making
3. Strategic release planning: Randomized versus deterministic
4. Operational release planning: Deterministic AND randomized
5. When-to-release decisions
6. Summary and outlook

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) Empirical research needed into the nature of decision-making
- More evidence for usefulness is needed