Software Release Decisions – Advanced Models and Optimization Methods

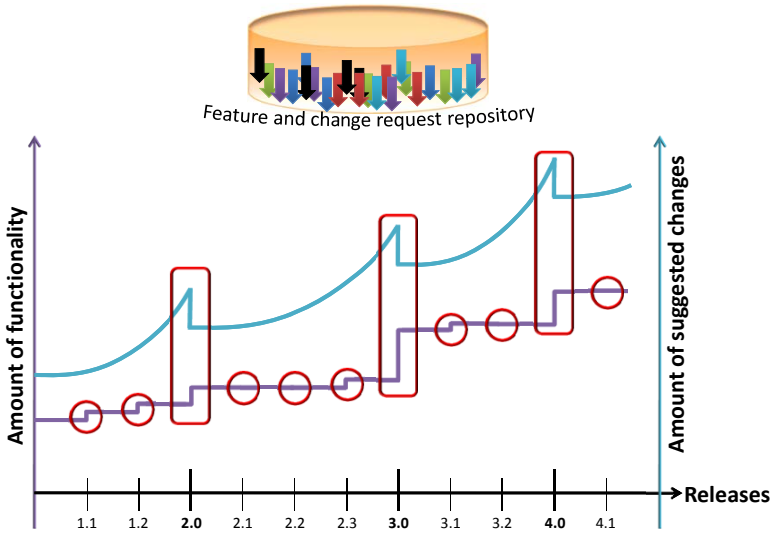
Guenther Ruhe

University of Calgary
Calgary, Alberta, Canada
ruhe@ucalgary.ca

Agenda

- Overview release decisions
- What-to-release
- What-to-release under complex feature dependency constraints
- When-to-release
- Re-planning
- Summary and outlook

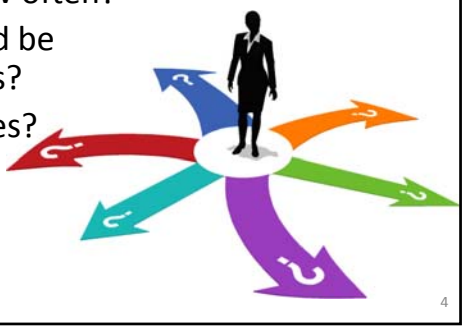
Release planning (RP) – What it is?



25th COW 3

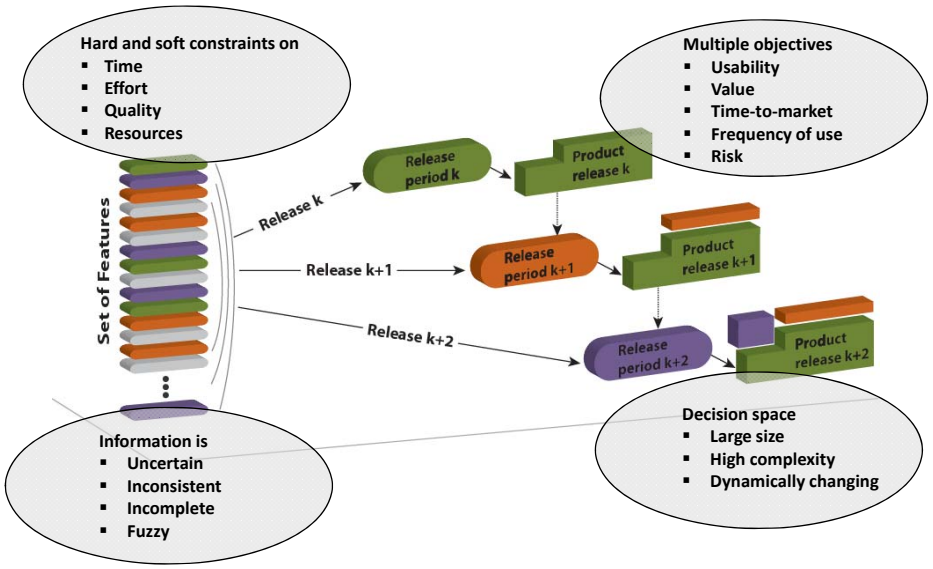
Release decisions

- Which features should be offered in the next release(s)?
- How to implement the next release (scheduling and staffing)?
- When is the best time for a product release?
- How to adjust to change for a given release?
  - When to re-plan? How often?
  - Which features should be replaced by new ones?
- How to plan for product lines?
- How to balance between quality and functionality?



25th COW 4

Release planning - Why it is difficult?



Agenda

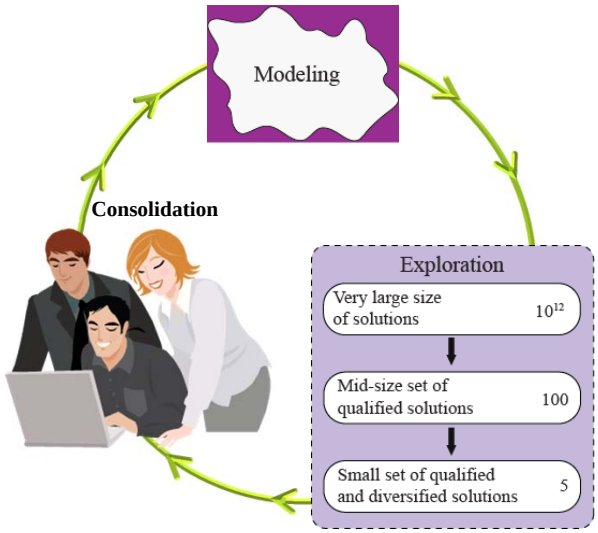
- Overview release decisions
- What-to-release
- What-to-release under complex feature dependency constraints
- When-to-release
- Re-planning
- Summary and outlook

The diversification principle

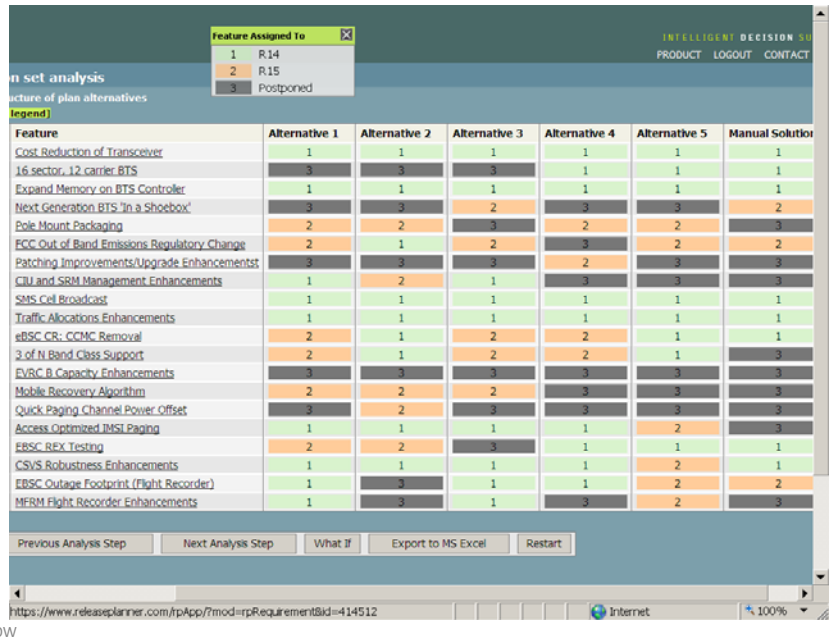
A single solution

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



Agenda

- Overview release decisions
- What-to-release
- What-to-release under complex feature dependency constraints
- When-to-release
- Re-planning
- Summary and outlook

Release planning by Constraint Programming (CP)

B. Regnell and K. Kuchcinski, "Exploring Software Product Management decision problems with constraint solving-opportunities for prioritization and release planning," in *Fifth International Workshop on Software Product Management (IWSPM)*, 2011, pp. 47-56.

Application of Constrain Programming:

MiniZinc: Syntax to define decision variables, input variables, and constraints using various logical and arithmetic operators such as

∧ (logical AND) ∨ (logical OR)

-> (logical implication)

= (equality) ≠ (inequality)

+ (addition) * (multiplication).

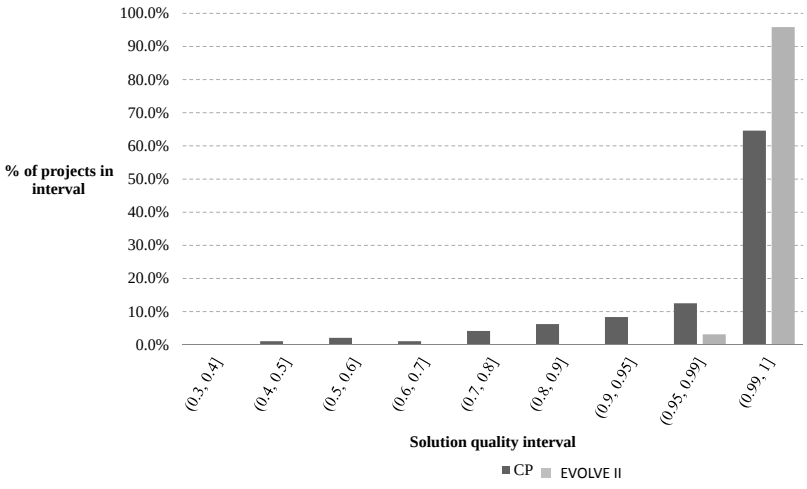
CP versus EVOLVE II: Efficiency[1)]

Tasks for 7 experimental subjects	CP (average)	EVOLVE II (average)	Unit
Adding a feature to feature repository	186	134	Seconds
Editing features	154	61	Seconds
Defining a dependency between features	135	102	Seconds
Defining available resources	147	73	Seconds
Editing available resources	197	61	Seconds
Inputting stakeholder votes	154	98	Seconds
Editing stakeholder importance	161	65	Seconds
Generating a solution	147	95	Seconds
Number of errors or failed commands	3	1	
Frequency of help or documentation use	6	1	

[1)] M. Przepiora, R. Karimpour, and G. Ruhe, "A hybrid release planning method and its empirical justification," Proc. *ACM/IEEE International Symposium on Empirical Software Engineering and Measurement ESEM 2012*, pp. 115-118.

CP versus EVOLVE II: Effectiveness

96 randomized projects. Defined by varying paramters N = {30, 150}, M = {1, 5}, K = {1, 5}, S = {N/25, N/10}, H = {N/10}, L = {N/50, N/10}, T = {0.2, 1.0, 3.0}.



Hybrid approach

Phase 1 (EVOLVE II)
If solution x* gained from EVOLVE II is feasible and there are no added value components in the objective function
then STOP **else goto** Phase 2

Phase 2 (CP versus Hybrid RP)
CP
2.1 Transformation of the problem using MiniZinc
2.2 Solution of the transformed problem using Gecode
Hybrid RP
2.1 Transformation of the problem using MiniZinc
2.2 Solution of the transformed problem using Gecode
2.3 Initiating solution process with x*

Empirical evaluation: Data sets

(1) Artificial data sets

Worst-performing project from the artificial dataset
- 150 features,
- 7 resources,
- 5 releases,
- 21 precedence constraints,
- 3 coupling constraints

Progressively addition of 0, 5, 10, 15, 20, 25, 30, 35, up to 40 non-EVOLVE II constraints.

(2) Real world data sets with added artificial constraints:

Project	# feat.	# res.	# releases	# dep.	# dep. features
1	25	7	3	23	25
2	75	5	2	81	75
3	633	1	5	19	67
4	914	1	4	85	232

Empirical evaluation: (Non-) Usage step 2.3

(1) Artificial data sets

# non-RPP constraints	Value (CP)	Value (Hybrid-RP)	Hybrid-RP/CP
0	20961	47843	2.28
5	21581	47874	2.22
10	23471	48173	2.05
15	22230	47477	2.13
20	22823	47802	2.09
25	21252	47148	2.21
30	21313	46533	2.18
35	22335	46533	2.08
40	24475	48299	1.97

Empirical evaluation: (Non-) Usage step 2.3

(2) Real world data set (ratio between values obtained from Hybrid-RP versus CP)

Project	after seconds		
	15	60	300
1	1	1	1
2	1.0132	1.01317	1.006
3	*	*	*
4	*	*	*

Agenda

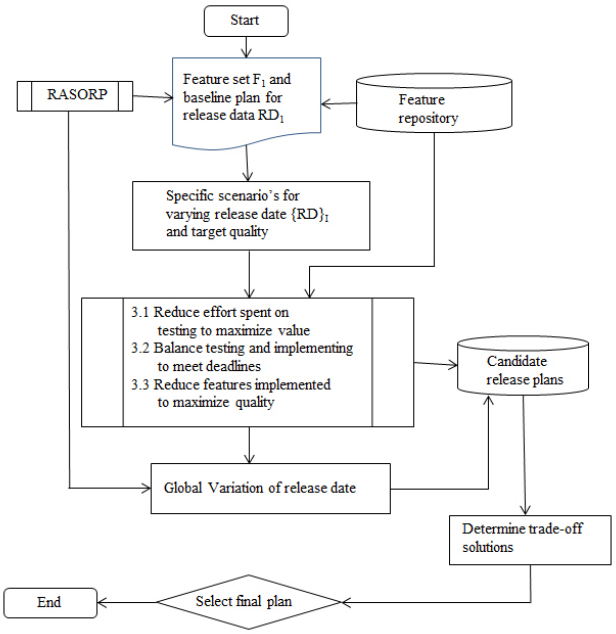
- Overview release decisions
- What-to-release
- What-to-release under complex feature dependency constraints
- When-to-release
- Re-planning
- Summary and outlook

When-to-release problem W2RP[1)]

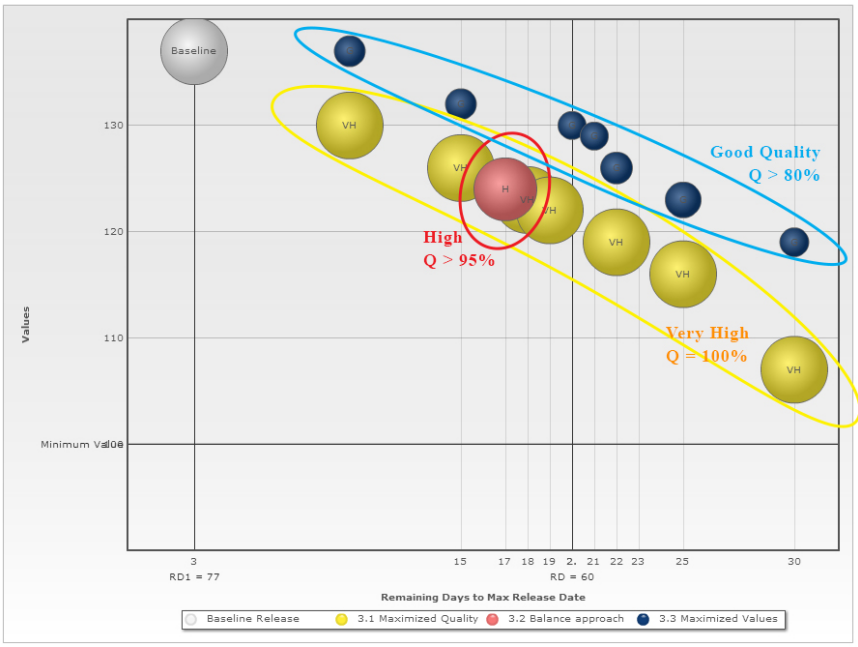
- Is defined by a sequence $\{RPP\}_i$ of problems RPP.
- Each individual RPP has a different fixed release date RD_i.
- W2RP means to determine operational release plans with varying feature sets F_i which represent trade-off solutions among all the variations of possible plans in terms of the three criteria
  - Maximize total release value $TRV(F_i)$
  - Maximize total release quality $TRQ(F_i)$
  - Minimize RD_i
- $CoQ(n) = CoC(n) + CLoC(n)$ for all features f(n)

[1)] J. Ho and G. Ruhe, "Releasing sooner or later? - An optimization approach and its case study evaluation," submitted to RELENG 2013

Re-optimization



Sets of trade-off solutions



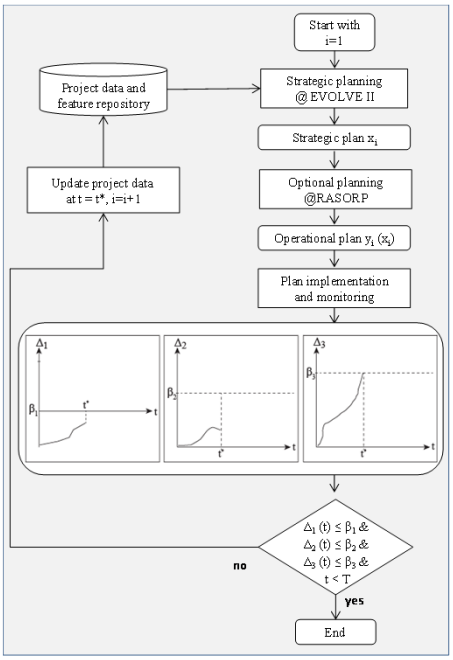
Agenda

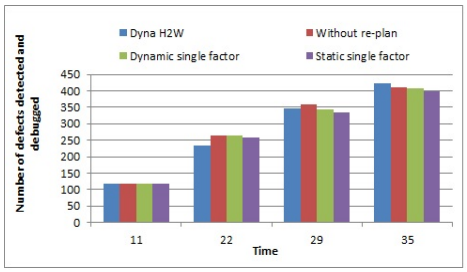
- Overview release decisions
- What-to-release
- What-to-release under complex feature dependency constraints
- When-to-release
- Re-planning
- Summary and outlook

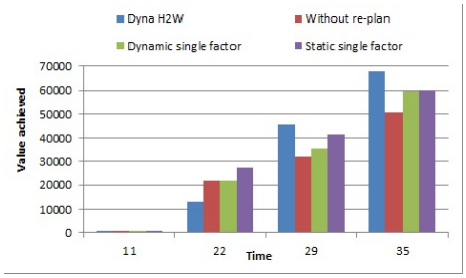
Dynamic Re-planning[1)]

- RQ1: How to perform dynamic re-planning of product releases in consideration of multiple factors?
- RQ2: How to integrate optimized operational and strategic planning into the re-planning process?

[1)] D. Al-Alam, G. Ruhe, and D. Pfahl, "Dynamic Re-planning of Software Product Releases - A Multi Factor Approach," submitted to RELENG 2013







25th COW 23

Agenda

- Overview release decisions
- What-to-release
- What-to-release under complex feature dependency constraints
- When-to-release
- Re-planning
- Summary and outlook

25th COW 24

Outlook

- Fundamental directions for improvement related to
  - Algorithms
  - Models
  - Empirical evaluation
- Planning for both functional and non-functional requirements
- New decision problems in the context of product lines
- Broader scope: More holistic planning including requirements elicitation, testing, design, project management

ssbse

Symposium on Search-Based Software Engineering, August 24-26, 2013

ESEC/FSE 2013
Saint Petersburg, Russia August 19–23

Important Dates

FULL PAPERS
Submission: April 5, 2013
Notification: May 17, 2013
Camera Ready: June 7, 2013

SHORT PAPERS/ GRADUATE TRACK PAPERS
Submission: May 3, 2013
Notification: May 27, 2013

Conference Dates: August 24-26, 2013