Quantitative release planning in Extreme Programming

Tommi Tervonen (1), Gert van Valkenhoef (1,2)
Bert de Brock (1), Douwe Postmus (2)

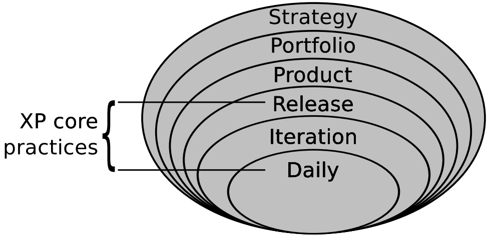
(1) Faculty of Economics and Business, University of Groningen (NL)
(2) Department of Epidemiology, University Medical Center Groningen (NL)

24th European Conference on Operational Research
July 2010, Lisbon

Introduction

- Traditional plan-driven software development methodologies (e.g. waterfall) cannot cope with changing user requirements, that are present in almost all projects
- Agile methodologies replace the strict plan-driven development process with values and practices proven to work well together
- Extreme Programming (XP) is one of the most agile software development methodologies
- The development in XP is guided by *user stories*, that are small pieces of visible functionality with added value for the customer

Release planning in XP



- The development team elicits user stories from the customer, who consequently prioritizes them
- Implementation complexity of stories are evaluated on the scale $\{1, 2, 3, 5, 8\}$
- Related stories can be grouped into themes that represent related functionality
- In each iteration, a velocity estimate amount of story points worth stories is selected for implementation

Problems in XP

1. **Customer availability**: the "whole team" practice requires constant presence of the customer
2. **Prioritization stress**:
    - in case of velocity change the customer might need to re-prioritize stories
    - customer might not perceive value in constantly prioritizing the stories

Our planning model

- We evaluate stories in addition to the implementation complexity with respect to their business value on scale $\{1, 2, 3, 4, 5\}$
- We incorporate themes to model synergy effects between stories. Theme valuation is difficult, as they have to be in the same scale with the story business values. Ordinal evaluation and value-free approaches (different functional forms) can be applied.
- We incorporate precedence constraints (e.g. story x needs to be completed before story y)
- We assume availability of a velocity distribution
- We produce “must have” (green), “should have” (yellow), and “could have” (red) lists

Cut-off points d_i for the green (b_1), yellow (b_2) and red (b_3) lists

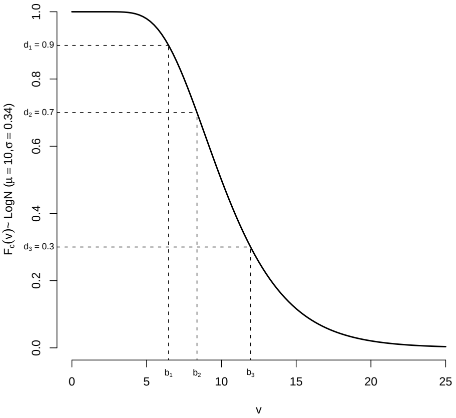


Figure: Complementary cumulative velocity distribution.

Let us define a set of stories $S = \{1, \dots, n\}$ and a set of themes $T = \{n+1, \dots, n+m\}$. All stories and themes have a business value u_i, and stories additionally have an implementation complexity c_i:

$$u_i \in \mathbb{N} \; ; i \in S \cup T$$
$$c_i \in \mathbb{N} \; ; i \in S$$

Define a nested set of knapsacks $K = \{1, \dots, \ell\}$ corresponding to the ℓ story lists, each with a discount factor (cut-off point) d_k and a budget b_k:

$$d_k \in \mathbb{R} \; ; k \in K$$
$$b_k \in \mathbb{N} \; ; k \in K$$

K is ordered according to the discount factors that satisfy:

$$d_i > d_j \; ; \; \forall_{i<j}$$

Define the decision variables of first including story s and theme t in knapsack k as $x_{s,k}$ and $y_{t,k}$, respectively:

$$x_{s,k} \in \{0,1\} \; ; s \in S, k \in K$$
$$y_{t,k} \in \{0,1\} \; ; t \in T, k \in K$$

Now, we optimize the following objective function:

$$\max \sum_{k \in K} \sum_{s \in S} x_{s,k} d_k u_s + \sum_{k \in K} \sum_{t \in T} y_{t,k} d_k u_t$$
$$\text{s.t.} \sum_{s \in S} \sum_{j=1}^{k} c_s x_{s,j} \leq b_k \quad \forall_{k \in K}$$
$$\text{and} \sum_{k \in K} x_{s,k} \leq 1 \quad \forall_{s \in S}$$

Completing themes is modeled through a dummy decision variable

$$z_{t,k} \in \{0,1\} \; ; t \in T, k \in K$$

that is true iff all stories in theme t are completed in knapsack k or any knapsack preceding k:

$$\left(\sum_{s \in S} \sum_{j=1}^{k} a_{s,t} x_{s,j} \right) - e_t z_{t,k} \geq 0 \quad ; \forall_{k \in K} \forall_{t \in T}$$

$$\left(\sum_{s \in S} \sum_{j=1}^{k} a_{s,t} x_{s,j} \right) - z_{t,k} \leq e_t - 1 \quad ; \forall_{k \in K} \forall_{t \in T}$$

Where $a_{s,t} = 1$ if story s is included in theme t and $a_{s,t} = 0$ otherwise, and $e_t = \sum_{s \in S} a_{s,t}$, the number of stories in theme t.

Then, we make sure that $y_{t,k}$ is true iff $z_{t,k}$ is the first (in terms of k) for which $z_{t,k} = 1$:

$$y_{t,1} = z_{t,1} \qquad \forall_{t \in T}$$
$$y_{t,k} = z_{t,k} - z_{t,k-1} \quad \forall_{t \in T} \forall_{k \in \{K-1\}}$$

The precedence relations, $i \prec j$ (i precedes j), are represented as follows:

$$x_{j,k} - \sum_{l=1}^{k} x_{i,l} \leq 0 \quad \forall_{i \prec j} \forall_{k \in K}$$

Velocity estimation heuristic: iteration

- If we have ≥ 5 velocity observations, the iteration velocity can be estimated through maximum likelyhood with

$$V_I \sim \log \mathcal{N}(\hat{\mu}, \hat{\sigma}^2)$$

where $\hat{\mu}$ is the mean of the log-transformed observations $\ln(\mathbf{v})$ and $\hat{\sigma}^2$ is the sample variance $\mathrm{sd}(\ln(\mathbf{v}))^2$.

Velocity estimation heuristic: release

- To estimate release velocity, release is viewed as a collection of n_R independent iterations. Release velocity is the sum of n_R log-normal distributions, and can be estimated using the (very accurate) Fenton-Wilkinson 2-moment approximation simplified for equal mean and variance:

$$V_R \sim \log\mathcal{N}(\mu_R, \sigma_R^2)$$
$$\sigma_R^2 \approx \ln(\exp(\hat{\sigma}^2) - 1 + n_R) - \ln n_R$$
$$\mu_R \approx \hat{\mu} + \ln n_R + \frac{1}{2}(\hat{\sigma}^2 - \sigma_R)$$

Velocity estimation heuristic

- Velocity estimate is overly precise in the beginning of a project, so we use the following weighted sum (approximating an inverse-Gamma prior with prior df=2:

$$\hat{\sigma} = \frac{\sigma_0 + n\,\mathrm{sd}(\ln(\mathbf{v}))}{1 + n}$$

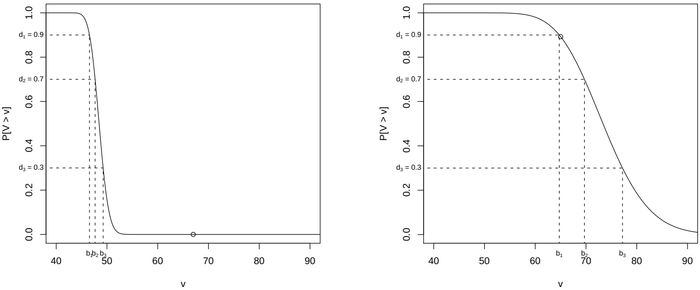
  where n is the number of observations and σ_0 is an prior belief of sample error that has a weight equal to one observation of true velocity.
- Prior belief σ_0 has to be specified!

Rules of thumb for uncertainty in velocity

Phase	Suggested CI	σ_0
Requirements Known *	$[\hat{\mu}/2.0, \hat{\mu} * 2.0]$	0.42
Requirements Analyzed *	$[\hat{\mu}/1.75, \hat{\mu} * 1.75]$	0.34
$<$ 2 Iterations Completed	$[\hat{\mu} * 0.60, \hat{\mu} * 1.60]$	0.29
Preliminary Design *	$[\hat{\mu}/1.40, \hat{\mu} * 1.40]$	0.21
Detailed Design *	$[\hat{\mu}/1.25, \hat{\mu} * 1.25]$	0.14
2 Iterations Completed	$[\hat{\mu} * 0.8, \hat{\mu} * 1.25]$	0.14
3 Iterations Completed	$[\hat{\mu} * 0.85, \hat{\mu} * 1.15]$	0.08
$>$ 3 Iterations Completed	$[\hat{\mu} * 0.90, \hat{\mu} * 1.10]$	0.06

With * are from NASA SEL guidelines (1990), others from Cohn (2005).

Velocity estimates vs observed velocity

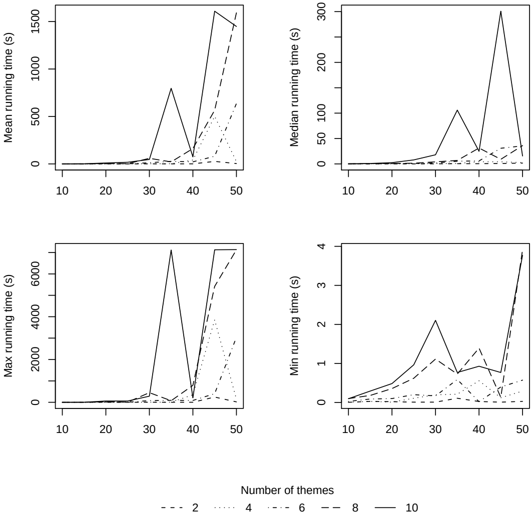


(a) V_R for R2 (based on $\mathbf{v}_1$)

(b) V_R for R3 (based on $\mathbf{v}_2$)

Figure: $F_c(v)$ estimated for release 2 (from release 1 velocity) and release 3 (from release 2 velocity). Due to higher variability during release 2, the estimated velocity is much less certain. The $\circ$ shows the velocity that was actually achieved.

Computational tests



Conclusions

- Release planning in XP can cause prioritization stress for the customer and is impractical in larger projects
- We developed an optimization model that enables XP for larger projects and for those with a less available customer
- The velocity distribution required for application of the model can be (easily) estimated with the provided heuristic, that corresponds well to velocity observed in a real-life development project
- Problems with up to 6 themes and 50 stories can be solved in less than an hour

Contact email: t.p.tervonen@rug.nl