Use case driven scheduling method of information technology project

T. GÓRSKI

gorski@wat.edu.pl, tomasz.gorski@rightsolution.pl

Institute of Information Systems
Cybernetics Faculty, Military University of Technology
Kaliskiego Str. 2, 00-908 Warsaw

In this article, requirements-oriented project scheduling method is presented. The method is use-case-oriented and shows how to prioritize use cases and schedule realization of information system. The method can be useful for project managers to plan and manage realization of information system.

---

1. Introduction

Managing requirements involves the translation of stakeholder requests into a set of system features. These in turn are detailed into specifications for functional and nonfunctional requirements. Detailed specifications are translated into test procedures, design, and user documentation.

According to Rational Unified Process [1] a requirement describes a condition or capability to which a system must conform, either derived directly from user needs, or stated in a contract, standard, specification, or other formally imposed document.

Requirements specify what the system must do rather than how the system does it. Requirements management is a systematic approach to:
- Eliciting requirements,
- Organizing requirements,
- Documenting requirements,
- Establishing and maintaining agreement between customer/user and the project team on the changing requirements.

A typical form of organizing requirements [8] in types and connecting them with traceability relationship was presented in the Picture 1.

Traceability allows you to determine the origins of any requirement. Software is expensive and difficult to develop. It is, therefore, important that you only deliver a solution that solves the business problem no more and no less.
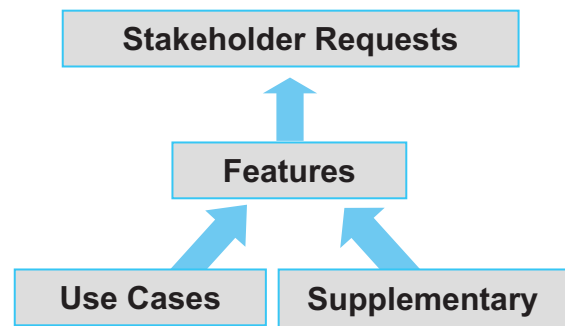
Picture 1. Typical way of organizing requirements

One of types of requirements is use case. Use cases represent functional requirements on a system. Use cases enable you to relate what you need from a system to how the system delivers on those needs.

A use case describes a sequence of interactions between actors and the system that occur when an actor uses the system to achieve a certain business goal [3].

A use case describes:
- The system, its environment, and the relationship between them.
- How things outside the system interact with the system.
- The desired behavior for the system.

Use cases are containers for contextually related requirements of the system under development. They are containers because they group all requirements related to achieving a particular goal into a single story of how that is achieved.

During requirements specification use case model is produced. Use case model consists of:

− Use case diagram,
− Use case specification document,
− Supplementary specification.

Because it is a very powerful planning instrument, the use-case model is generally used in all phases of the development cycle by all team members. Use cases are used to plan content of iteration. An example of use cases allocation to iterations was depicted in Picture 2.
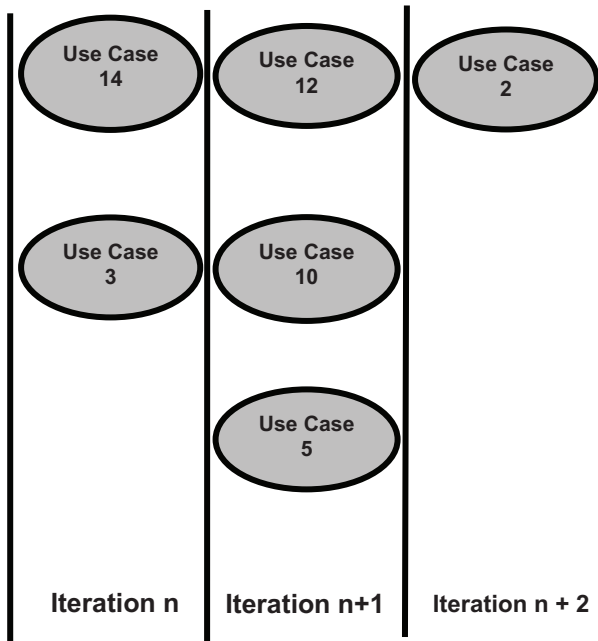
Picture 2. An example of use cases allocation to iterations

Benefits of use cases [8]:

− Give context for requirements.
  ◆ Put system requirements in logical sequences.
  ◆ Illustrate why the system is needed.
  ◆ Help verify that all requirements are captured.
− Are easy to understand.
  ◆ Use terminology that customers and users understand.
  ◆ Tell concrete stories of system use.
  ◆ Verify stakeholder understanding.
− Facilitate agreement with customers.
− Facilitate reuse: test, documentation, and design.

As presented in Picture 2 use cases are used to plan iterations. Scheduling method of information technology project proposed in that paper shows how to allocate use cases to specific iterations.

2. The scheduling method

Use cases are described by set of attributes. Attributes which describe an use case could be divided into two groups:

− Describing qualities of use case,
− Used to use case management.

Attributes describing qualities of use case could be following:

− **Benefit** – use case importance for stakeholders. Values: Critical, Important, Useful.
− **Effort** – could be expressed in terms of number of work weeks of development team.
− **Technical risk** – realization risk connected with used technologies. Values: High, Medium, Low.
− **Stability** – describes level of volatility of requirement. Values: High, Medium, Low.
− **Cost** – costs connected with use case realization which can be expressed in money.

Attributes used to use case management:

− **Status** – formal state of requirement. Values: proposed, approved and incorporated.
− **Planned iteration** – stores number of planned iteration in which specific use case should be incorporated.
− **Number in iteration** – stores number of use case in planned iteration sequence,
− **Origin** – this attribute is used to store information about feature which is realized by this use case.
− **Assigned to** – a person or team who is responsible for use case realization.

Attributes of use cases are usually presented in form of attribute matrix. In the picture 3 attribute matrix was depicted for several use cases.

	Status	Risk	Priority	Effort	Cost
UC 10	Approved	Low	High	3 weeks	40000
UC 13	Proposed	Medium	Low	1 week	15000
UC 40	Approved	High	High	2 weeks	25000

Picture 3. An example of attribute matrix for use cases

In order to apply the method, values of attributes should be numbers from the range <0,1>. For attributes whose values are not numbers were set numerical values which correspond to textual ones.

Following values were set for attributes:

Benefit:
- Critical – 0.9,
- Important – 0.5,
- Useful – 0.2,

Technical risk:
- High – 0.9,
- Medium – 0.5,
- Low – 0.2,

Stability:
- High – 0.9,
- Medium – 0.5,
- Low – 0.2.

In the method certain rule was applied: for attributes with highest importance (Critical, High) was established numerical value 0.9, for attributes with middle importance (Important, Medium) was set numerical value 0.5 and for attributes with the lowest importance (Useful, Low) was defined numerical value 0.2.

Below example of attribute matrix with numerical values of attributes was presented.

	Risk	Stability	Effort	Cost
UC10	0.2	0.9	3	40000
UC13	0.5	0.2	1	15000
UC18	0.5	0.9	2	22000
UC27	0.2	0.2	4	43000
UC39	0.2	0.9	4	45000
UC40	0.9	0.9	2	25000

Picture 4. An example of attribute matrix for use cases with numerical attributes values

Hardly ever, do we have situation when a project manager depends on value of one attribute. More common case is when many factors have impact on project manager's decisions.

A project manager is definitely interested in:
- Quickest delivery of maximum system's functionality which is crucial for customer,
- Minimizing effort, technical risk and costs.

It is also important that the most stable use cases should be realize first. Hence, if we want to schedule use cases then we have to use many criterions of selection. In that case we deal with multi-criterion optimization.

Among multi-criterion optimization methods can be enumerated [6]:
- Pareto optimization,
- Mini-max methods,
- Method of consecutive resignations,
- Coefficients' importance method.

In proposed scheduling method of information technology project, coefficients' importance method was applied. Below was presented proposed scheduling method with application of selected optimization method to allocate use case to iterations.

First we should define vector of criterions (attributes which describe use case qualities), where M is number of attributes.

$$O = < a_1, a_2, ..., a_m, ..., a_M >, m = \overline{1, M}, \quad (1)$$

Furthermore, we should identify set of a project use cases P which consists of elements $p_k \in P, k = \overline{1, K}$, where K is number of use cases.

$$p_k = < a_1, a_2, ..., a_m, ..., a_M >, k \geq 2, M \geq 2, \quad (2)$$

We have to normalize values of all criterions a_m to normalized values a_m^N within range <0,1>

$$a_m^N = \frac{a_m - a_{m \min}}{a_{m \max} - a_{m \min}}, \quad (3)$$

Considered normalization of criterions' values a_m is indispensable so as to all criterions have identical domain.

Next, we have to determine importance coefficients f_m for each criterion

$$< f_1, f_2, ..., f_m, ..., f_M >, where \sum_{i=1}^{M} f_i = 1, \quad (4)$$

Importance coefficients for individual criterions are set subjectively. Those coefficients define importance degree of criterion participation in final result. In order to present results in percents, sum of all coefficients must be equal 1 and normalized range of criterions' values must be <0,1>.

In the following step, value of factor A_m should be determined for each criterion. Factor A_m defines criterion's influence on use case assessment. This factor can increase value of assessment (maximizes an assessment) or decrease value of assessment (minimize an assessment).

Factor A_m can have following values:

1 – for criterions increasing value of assessment,

-1 – for criterions decreasing value of assessment.

In the next step, we should calculate (final result) value of assessment for use case w_k.

$$w_k = \sum_{i=1}^{M} A_i \cdot f_i \cdot a_i^N \,, \qquad (5)$$

This should be done for each use case so as a result we will get vector of values of assessment w_k for all use cases.

$$W =< w_1, w_2, ..., w_k, ..., w_K >, k = \overline{1, K} \,, \quad (6)$$

Then we have to apply following function to schedule use cases.

$$\max(w_k) \,, \qquad (7)$$

As a result we will get vector of scheduled use cases according to applied function.

$$VP =< p_1^u, p_2^u, ..., p_k^u, ..., p_K^u >, k = \overline{1, K} \,, \quad (8)$$

Next step is allocation of scheduled use cases from vector VP to vectors of consecutive iterations VIm. Further, an example of vectors VIm was depicted (9).

$$VI_1 = \left\langle p_1^u, p_2^u, p_3^u \right\rangle, VI_2 = \left\langle p_4^u, p_5^u \right\rangle, ... VI_m = \left\langle p_k^u, ... p_K^u \right\rangle, \quad (9)$$

The allocation should be done with consideration of duration time of iteration. Duration time of each use case is set by expert and stored in attribute Effort. For each vector VIm sum of values attribute Effort for All use cases in iteration m should not exceed duration time of iteration.

3. Design of scheduling application

Design of scheduling application was realized according to Rational Unified Process methodology. IBM Rational Software Modeller was used to model requirements and design of the application. All models were presented in Unified Modelling Language 2.0. Whole design of the application was divided into two elements: Use-Case Model and Design Model.

First of all Use-Case Model was created. This model served as a definition of scope of system's functionality. In the use case diagram actors and application's use cases were depicted. Use case diagram of designed applications with one use case "Schedule project" (pol. Harmonogramuj projekt) was presented in the picture 5.

Picture 5. Use case diagram of designed application

The complete flow of events of "Schedule project" use case was depicted in activity diagram of UML 2.0. The complete flow of events of "Schedule project" use case was presented in the picture 6.

Picture 6. Activity diagram of "Schedule project" use case

Furthermore design model was created. Static structure of software was described on class diagrams and dynamic structure of software was presented on sequence and communication diagrams.

In the picture 7 class diagram of scheduling application was presented.

Picture 7. Class diagram of designed application

Next operations required to realize scheduling task should be identified. In order to do that sequence and communication diagrams were used. Sequence diagram which presents realization of scheduling task by designed classes is shown in the following sequence diagram (Picture 8).

Picture 8. Sequence diagram of scheduling application

As a next step communication diagrams were generated from designed sequence diagrams. Communication diagram which presents realization of scheduling task by designed classes is shown in the following communication diagram (Picture 9).

Picture 9. Communication diagram of scheduling application

Pattern of communications (relationships) is clearly visible on communication diagrams. This diagram was used to refine class diagram, previously presented in picture 7.

The next and last step was implementation of operations in refined classes.

4. The scheduling method implementation

Implementation of the method was realized in IBM Rational RequisitePro – tool designed to requirements management. IBM Rational RequisitePro is integrated with Microsoft Project. Microsoft Project Intergration Wizard connects requirements in IBM RequisitePro with tasks in Microsoft Project

Specific add-in was developed and added to Microsoft Project. This add-in is invoked from the option in Microsoft Project menu (Picture 10).

Picture 10. Added scheduling functionality to Microsoft Project

In that add-in whole logic of scheduling method was designed and developed. Data about use cases are retrieved from IBM Rational RequisitePro by means of API RequisitePro Extensibility Interface (RPX) which is part of designed add-in. This add-in allows project manager to select use cases' attributes to schedule a project (Picture 11).
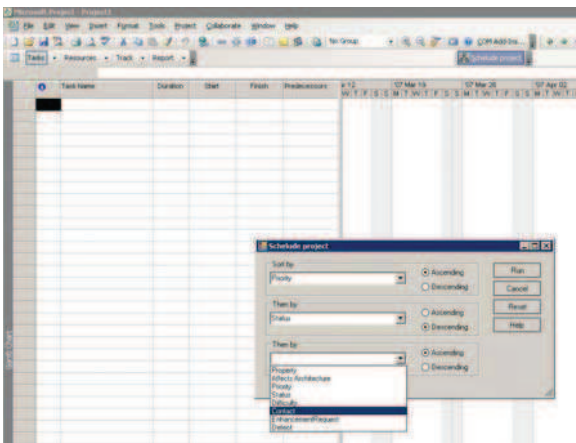
Picture 11. Form for attributes selection to schedule a project in MS Project

As a subject of allocation are only considered use cases which have attribute Status set to value "Approved". In that moment, task of multi-criterion optimization is run. As a result we will get vector of scheduled use cases VP.

Next, taking into consideration a duration time of iteration Tit and values of attribute Effort, allocation of use cases from VP to specific iterations is being done (vectors VIm). As a result, values of two attributes, for each considered use case, are changed: Planned iteration and Number in iteration.

Scheduled use cases are shown in Microsoft Project in the form of tasks' list and Gantt's chart. This kind of solution does not require

opening IBM Rational RequisitePro which requires using of license.

Microsoft Project and scheduling function could also be started from IBM Rational RequisitePro by using, described above, add-in (Picture 12).
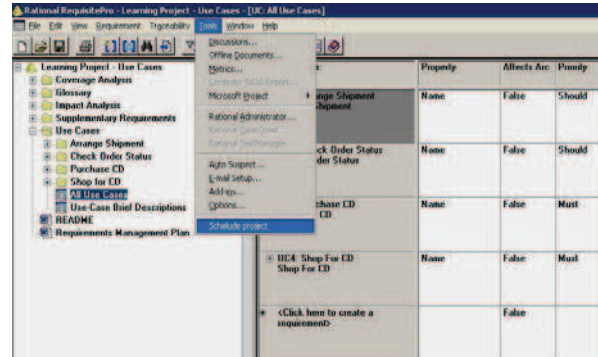
Picture 12. Added scheduling functionality to IBM Rational RequisitePro

So, designed add-in allows selection of use cases' attributes to sort use cases and their allocation to iterations also from IBM Rational RequisitePro environment (Picture 13).
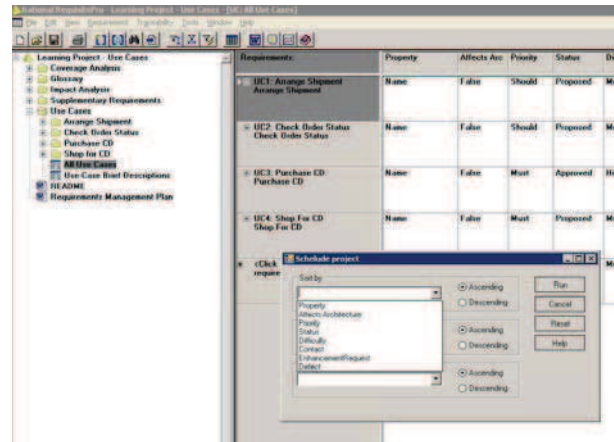
Picture 13. Form for attributes selection to schedule a project in RequisitePro

As a result of scheduling a project are filled values of attributes: Planned iteration and Number in iteration. Values of these attributes are stored in IBM Rational RequisitePro project. It should be emphasized that the method takes into consideration only those use cases for which value of attribute Status is set to "Approved".

In the Picture 14 attribute matrix of example project with scheduled use cases was presented. Attributes filled by scheduling application were encircled with dashed rectangle.

Picture 14. Attributes filled by scheduling application

Furthermore, designed add-in creates Gantt diagram in Microsoft Project. This diagram is based on use cases and values of attributes Planned iteration and Number in iteration.

In the picture 15 example of Gantt diagram created by scheduling application was presented.
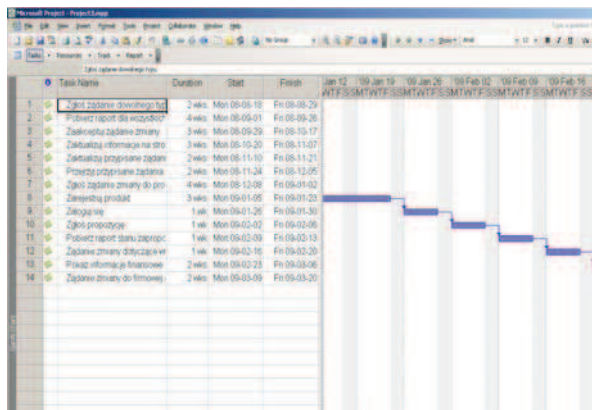
Picture 15. Gantt diagram for example project

5. Summary

This kind of scheduling method, implemented in commercially available tool, appears to be one of the most desired supporting tools for project managers. It gives them content-related input for project planning. Moreover project manager is able to benefit from that method during the whole project. It is common knowledge that intrinsic part of information technology project is change. In case of change, thanks to proposed method, a project manager can reassess values of attributes and dynamically change schedule of specific project.

It is worth mentioning that due to simplicity of the method implemented application operates very fast and generates Gantt diagram within few seconds.

6. References

[1] Kruchten P. – *Rational Unified Process - the introduction*, Addison-Wesley, 1999, ISBN 0201604590

[2] Eeles P., Houston K., Kozaczyński W. – *Building J2EE applications with the Rational Unified Process*, Addison-Wesley, 2003, ISBN 0-201-79166-8

[3] Cockburn A. – *Writing effective use cases*, Addison-Wesley, 2001

[4] Leffingwell D., Widrig D. – *Managing software requirements. A unified approach.*, Addison-Wesley, 2000

[5] Kruchten P. – *What is the Rational Unified Process?*, 2003

[6] Roy B. – *Wielokryterialne wspomaganie decyzji* (Multicriterial decision aid), WNT, Warszawa 1990, ISBN 83-204-1119-X

[7] Górski T. – "Requirement-driven software engineering process", V Polish Conference of IBM Rational Technology Users, Software People Engineering, 22nd April 2005, Warsaw

[8] *Mastering of requirements management with use cases*, IBM Materials, 2003

[9] Górski T. – "Scheduling method of information technology project", International Conference, Information Management, IM/HCI 2007, Jurata, 12- 14 June, 2007.