M-applications Development using High Performance Project Management Techniques

PAUL POCATILU, MARIUS VETRICI
Economic Informatics Department
Academy of Economic Studies
6 Piata Romana, Sector 1, Bucharest
ROMANIA
ppaul@ase.ro, mariusvetrici@softmentor.ro

Abstract: This paper analyses the specifics of m-applications development projects. Even though m-application development projects have their particular features and need customized approaches, still we find that legacy and proven best practices project management techniques can be successfully employed. We present two proven software project management techniques that were successfully adapted to the development of m-applications. One is the estimation of m-application project duration using top-down and bottom-up approaches. The other is the use of a set of performance metrics for project quality assessment.

Key-Words: Mobile applications, Mobile devices, Project management, Performance metrics

1 Introduction
The total number of mobile phone subscribers in the world was estimated at 2.14 billion in 2005 [1], 3.3 billion in 2007 [2] and the figure is expected to increase to 90% by the year 2010. The numbers are even more impressive if we look at the mobile phone penetration rates, the highest from Asia being in Hong Kong with 1.4 mobile phones per person [3] and in Europe, Luxembourg, Lithuania and Italy hitting as high as 150 mobile phone subscriptions per 100 people [4]. Given the circumstances, m-application software development is and will be an emerging field of the software industry.

An m-application is a special type of software application particularly designed to be used on mobile processing units with limited processing power, storage memory and input capabilities such as mobile phones, smartphones, PDAs, navigation assistants, mobile guides, etc. Even though m-application development is a relatively new domain, still legacy project management techniques can be successfully applied for delivering high performance in the new field. This paper analyses the specifics of the m-applications development projects and presents two proven software project management techniques that can be successfully adapted to the development of m-applications.

First technique, the estimation of project duration, is of utmost importance for all project stakeholders. More specifically, the duration of the project is needed before the project has started. This is because other important estimations are grounded on the former metric. For example no investor will go with a given project unless the delivery date is clearly agreed upon and a commitment has been entered into. Further to this, we go into great detail about the importance of project duration estimation, the difficulties of estimating duration and the existing duration estimation techniques.

The second project management technique relies on the use of specific m-applications development performance metrics. The performance metrics are based on customer satisfaction, the degree of objective completion and the cost of the resources involved.

2 M-applications Development Projects
M-application development is similar to the development of personal computer applications, but there are also differences that influence the way the project is managed. Depending on the application type, m-applications development projects include not only the mobile device software, but also the other components of the system (application server, database, content management etc.). We will focus mostly on the development of the mobile applications.

From the data processing point of view, m-applications can be divided into standalone applications and distributed applications.

Standalone mobile applications are designed to perform specific tasks without the need of a network connection. Mostly mobile applications made for PDAs are such examples of stand-alone applications.

Every operating system (Windows Mobile, Symbian) exposes specific APIs with varying degrees of complexity and architectures which are more or less well documented. In order to increase the development productivity, higher level classes libraries were developed on top of system’s APIs. Usually, every
library comes with a specific run-time environment.

Distributed applications instead need a network connection in order to operate. This type of applications may rely upon a permanent or a temporary connection. WAP (Wireless Access Protocol) based applications for mobile phones that connect to a server via Internet are an example of distributed applications. The most used distributed applications are Web-based. Figure 1 depicts the architecture of such an application.

The request from the WAP enabled phone is sent to the WAP gateway that makes the conversion from the WAP stack (for WAP 1.0) or from the optimized wireless or optimized HTTP/TCP/IP (WAP 2.0) to the HTTP/TCP/IP stack and encodes the network packets that will further be sent to the Web server as an HTTP request. The request is processed by the Web server, and then a response is send back to the mobile phone browser through the WAP gateway that decodes the packets.

The type of application has an important influence on the size and the complexity of the m-application development project:

<table>
<thead>
<tr>
<th>Application Type/Features</th>
<th>User Interface</th>
<th>Memory</th>
<th>Processing Power</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network access</td>
<td>Limited</td>
<td>High</td>
<td>Medium /High</td>
<td>High</td>
</tr>
<tr>
<td>Stand-alone</td>
<td>Limited</td>
<td>Medium /High</td>
<td>Medium /High</td>
<td>Medium /High</td>
</tr>
<tr>
<td>Web-based</td>
<td>Web-based</td>
<td>Medium /High</td>
<td>Low /Medium</td>
<td>Low</td>
</tr>
<tr>
<td>Database access</td>
<td>Limited</td>
<td>High</td>
<td>Medium /High</td>
<td>High</td>
</tr>
</tbody>
</table>

As it can be seen from the table 1 [5], mobile applications that require network access and those that use databases usually have a higher complexity. This is rather obvious because this type of m-applications will have greater complexity, more classes and chances are that the demand for specific knowledge will be higher.

The size, complexity and productivity are influenced by the application’s operating system. Using Java ME technology there is a high degree of portability between operating systems, but there are device specific influences.

The use of native APIs to write applications requires more effort, and the size of application (expressed as KLOC) is higher than using classes libraries.

Most of the development process is made using device software emulators that run on personal computers. Still there are differences between real-life devices and emulators. That’s why there is an additional effort in testing the application even after it is considered done on the emulator.

3 Time/Duration Management Models

3.1 Definitions
A project is “a temporary endeavor undertaken to create a unique product, service, or result” [6]. By adapting the definition from [7] we state that an m-application software development project is a temporary endeavour undertaken to create a unique m-application. High quality m-application software development projects deliver the required product within scope, on time and within budget. It is the project manager’s duty to skilfully balance the competing demands for project quality, project duration and cost of resources in order to be able to deliver the software as planned.

Like any other type of project, software development projects need:

- clearly defined requirements and scope
- established achievable objectives
- controlled resource allocation
- good effort and schedule management

The expectations of stakeholders are focused on the software to be delivered, on the budgeted consumption and on the project duration.

The duration of a project is the time elapsed between the project start and the project delivery date, when the software is delivered to the customer. The project duration is an essential indicator that should be well estimated, agreed upon with the stakeholders and thoroughly monitored, up to the project completion.

Project duration and size reflect the manager’s own understanding of the requirements. It is not possible to correctly size and estimate duration for a project that is not completely understood. Further, project duration provides an important check for scope creep throughout the project. Failing to pay attention to project duration one could agree to add new functionality without appropriately updating project size and effort needed.

3.2 The difficulties of estimating software project duration
There are several reasons that make m-application project duration estimation a difficult problem. First of
all, the very essence of software building process makes it difficult to measure. It is a tough endeavour to try to measure “how much” software is there in a software project because the software is invisible and unvisualizable [8]. This especially difficult if we try to make such forecasts before a detailed software design.

The software is pure thought-stuff, infinitely malleable [8]. Unlike cars and buildings, the software is constantly subject to pressures for change because the costs of modifications are difficult to understand.

Many of the classic problems of developing software products derive from this essential complexity and its nonlinear increases with size. From the complexity comes the difficulty of communication among team members, which leads to product flaws, cost overruns and schedule delays. From the complexity comes the difficulty of enumerating, much less understanding, all the possible states of the program, and from that comes the unreliability [9].

3.3 Duration estimation techniques

The grand majority of techniques for m-application project development duration estimation can be found either in bottom-up or top-down category. The difference between the two comes from the approach used to estimate project duration. The techniques in the first category start at the task-level view of the project and aggregate the work to be performed on higher levels, up to the project as a whole. The top-down way offers duration predictions based on properties of the work-product, the project team, and the project environment, figure 2.

3.3.1 Bottom-up techniques

This type of duration estimation techniques start with developing a work breakdown structure of the work and then continue with task identification and task duration estimation. Every task should be simple enough so as one could easily answer the question regarding the task duration three parameter estimates:

- best duration estimation
- most likely
- worst duration

Also for every task one should know:

- what's involved in getting started
- how will resources be allocated
- what exactly are the conditions to be met in order the project to be considered done.

The next step is identifying the predecessor-successor relationships and the critical path through the activity graph.

In order to forecast the completion time, three different approaches can be used.

a) The simple approach consists in adding-up the most likely estimates for each task on the critical path. It is not the best method, but it is the simplest one.

b) The second approach means to calculate the expected task duration ED as a weighted mean of the three given estimations using PERT equation:

\[
ED = \frac{BD + 4 \times MD + WD}{6}
\]

where:

BD – best duration estimation; this is the most optimistic expectation, the best case scenario that assumes no influence is going to negatively impact the project duration;

MD – most likely; the duration of activity given the resources, their productivity and realistic expectations of availability;

WD – worst duration; the duration of activity based on a worst case scenario of what is described in most likely estimate.

c) The third approach relies on a Monte Carlo simulation over the task estimation data. The result will be a probabilistic distribution of the project duration [10]

3.3.2 Top-down techniques

Top-down techniques use instead some high level attributes of the project (related to its complexity, functionality or size) and of the organization capability to deliver the project.

Top-down estimation begins with an assessment of the size of the work-product being planned. This idea comes from the construction projects, where the project-manager wouldn't imagine committing to a deadline without establishing and tracking some good size estimates, like the number of square feet, number of windows, doors, etc. to be designed and built.

Up to date there are four software project sizing legacy methods. See table 2 [11]:

<table>
<thead>
<tr>
<th>Sizing Method</th>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lines of Codes</td>
<td>Easy to measure</td>
<td>Cannot be done</td>
</tr>
</tbody>
</table>
### Function Points

<table>
<thead>
<tr>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Can be measured during requirements stage.</td>
<td>Requires some training, calibration and perhaps tailoring to specific application domains.</td>
</tr>
</tbody>
</table>

### Use-Case Counting

<table>
<thead>
<tr>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Can be measured during requirements stage.</td>
<td>New method. Small experience base at this time.</td>
</tr>
</tbody>
</table>

### Web Application Proxies

<table>
<thead>
<tr>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Easy to count starting with early web application prototypes.</td>
<td>New method. Requires development of counting rules and calibration for specific application types.</td>
</tr>
</tbody>
</table>

The next step in top-down estimation is to use a project duration estimation model.

Lawrence Putnam proposed a widely used model for project duration estimation using data on size, effort, and historic duration for thousands of other software projects. The model builds up the organization's delivery capability index using PP - Productivity Parameter and links it to size, effort and duration dynamics.

\[
PP = \frac{PS}{\left(\frac{E}{\beta}\right)^{1/3} \ast D^{4/3}}
\]

where:
- PP – Putnam’s productivity index. This item shows the organization’s project delivery capability;
- PS – project size, counted using one of the above sizing methods;
- E – effort (in man-years). The work needed in order to fulfill the project;
- D – the project duration (years).

The following things are notable in regard to this model:
- an organization with higher PP can deliver more size with less effort and in shorter duration than one with a lower PP;
- the 1/3 and 4/3 exponents in equation 2 express the non-linearity in effort-duration relationship.

### 3.4 Choosing a project duration estimation technique

Both top-down and bottom-up approaches proved to be good at estimating project duration. A good software project manager will probably use both methods, plus his own estimation, based on priori experience. Bottom-up estimates use work-breakdown structure, critical path method and task estimates; they provide crucial details regarding the duration of smaller project parts and they roll up to a global duration and effort estimation. Top-down estimates rely on history of other real projects. One's cumulative experience in similar projects can provide estimates that deserve some consideration in balance with the bottom-up and top-down views.

### 4 M-applications Development Performance Metrics

Poor project management is the number one factor of the IT projects failure, including m-applications development. Upon completion, a project can meet all the objectives and still it can be a financially unprofitable project.

High quality project deliverables cannot be obtained without high quality development processes, but a quality process does not guarantee quality products. The quality of the process is certified through quality standards.

Also good trained personnel do not guarantee the quality of deliverables. In order to obtain quality results, the organization has to have trained and educated personnel, and standardized project management and technological processes.

Equilibrium has to be obtained between: resource allocation for projects, risk and profit, long term and short term projects, research and development projects, internal or external projects. In figure 3 is depicted a situation from an organization with some projects showing the associated risks, value and profit. In [12]...
several indicators were proposed for IT project performance measurement. These indicators can be applied to measure the performance of m-applications development projects.

The degree of objectives achievement is calculated as:

\[ GA = \frac{OA}{TO} \]  

(3)

where:

OA – the number of achieved objectives
TO – the total number of established objectives

If the indicator value is greater than one, is considered that the project achieved more objectives than were planned initially.

The ratio between the achieved deliverables and the planned deliverables can be also calculated for each project phase, where deliverables from one phase are inputs for the next phase.

The degree of satisfaction can be computed as:

\[ DS = \frac{\sum_{i=1}^{p} DSR_i}{TR} \]  

(4)

where:

DSR – the degree of satisfaction for the requirement i
TR – total number of requirements
p – the number of requirements

The degree of satisfaction for a customer requirements is a value from 0 (no satisfaction) to 1 (fully satisfied) or using a similar scale. The degree of client satisfaction with an m-application can vary with the mobile devices the application is run on.

Work productivity based on inputs is given by:

\[ W_1 = \frac{\sum_{i=1}^{n} O_i}{\sum_{j=1}^{m} I_j} \]  

(5)

where:

O_i – the output i; (deliverables, results)
I_j – the input j (work, resources per time unit)
n – the number of outputs
m – the number of inputs

Work productivity based on time:

\[ W_2 = \frac{\sum_{i=1}^{a} O_i}{T} \]  

(6)

where:

T – period of time

The cost of resources takes into account the category of resources and the cost per unit for each category:

\[ C = \sum_{i=1}^{m} NR_i d_i p_i \]  

(7)

where:

NR_i – number of resource from the category i
pi – price per unit for the resource category i
di – units of usage for the resource category i

The total cost of a project can be defined as:

\[ C_T = \sum_{i=1}^{k} c_i \]  

(8)

where

k – the number of project phases

Other indicators are developed to measure the performances of IT projects, having in mind the m-applications characteristics. In order to use them, data have to be collected from various projects and they have to be validated [13].

5 Conclusions and Future Work

M-application software development is an emerging field of the software industry. Despite being a relatively new field, best practices project management techniques can be successfully used to deliver high performance.

The development of mobile applications involves some difficulties engendered by reduced capabilities of mobile devices. Due to mobile devices limitations, in particular limited internal memory and reduced processing power, the source code of mobile applications needs additional optimization which will result in less testability.
M-application project development implies the usage of specific development environments like emulators that are not 100% compatible with the hardware device. This difference requires a slightly different approach both for development and testing.

M-application project duration can be successfully estimated using top-down and bottom-up approaches that have successfully been used over the last decades.

In order for the m-application to be evaluated as successful, a quantitative approach can be employed by the use of a set of performance metrics.

Further research will be focusing on the use of prepackaged m-components as a means of speeding up the development process. Also, m-application project development success will be measured by assessing the quality of the m-application user interface.

References:
[3] ***, Key Telecommunications Statistics, Office of the Telecommunications Authority in Hong Kong