**1)** **Identify the hardware support your team will need to support development of software to run on a Windows CE handheld device. Be sure to consider development support and testing support.**

**Solution:**

For Windows CE 2013, the development hardware needs to be able to host virtual test device. The prerequisites to host a virtual test device and create a single virtual machine for a test device are as following [1]:

* At least 4 GB of RAM. Out team need addition RAM to run several virtual machines.
* 15 GB free hard drive space for each virtual machine.
* A TCP/IP network connection with access to a DHCP server. Because the virtual machine uses a virtual Ethernet switch, we can use any physical interface for this connection.
* The computer may have a 64-bit processor that supports Second Level Address Translation (SLAT).

Some popular workstations that meet with above requirements include: ThinkPad X1 Carbon, Dell XPS 27 AIO, Microsoft Surface Studio, etc.

Other development hardware includes:

* Network adapter (e.g. Asus, Cisco or Linksys wireless adapter).
* DHCP server. (HP, Cisco, etc.)

For Windows CE 2013, the testing hardware support must meet the following minimum hardware requirements [1]:

* 100 GB of available hard disk space
* 1.6 GHz processor (2.4 GHz recommended)
* 2 GB of RAM
* 5400 RPM hard drive
* DirectX 9-capable video card running at 1024 x 768 or higher display resolution

Some cheaper and more affordable workstation can be used as testing hardware. For example, Dell Precision 3630, ThinkStation P30, etc.

**2)** **Consider a project you were involved with in the recent past. Reconstruct a series of project scoreboards covering a six- to eight-week period. Draw some conclusions from these scoreboards, given the success level of the project.**

**Solution:**

As stated in textbook, a project scoreboard is effective for seeing and understanding current project status. In order to make measurements a regular part of project use a project scoreboard, where the team records information in a form that shows schedule impact, commitments, dependencies, risks, decision, and important project comments [2]. The scoreboard often uncovers potential problems and allows analysis of the extent and complexity of known problems.

A scoreboard’s function is to provide an overview of the project and its progress and status. It should include several metrics for the project, but only those are suited to or relevant to the project count [3]. The core of a project scoreboard is the project timeline, every project has a timeline, the stakeholders and leadership should see from the scoreboard the project’s milestones, phases across the timeline, and how the project is progressing against the schedule [4]. Another important fact of scoreboard is, measurement can provide quantitative information on what has been produced or accomplished. Since measurement is a useful monitoring tool it should be gathered at regular intervals during the project, and project manager should make sure the effective data be acquired and populate the measurements.

Table 1 is one project scoreboard; its scope is project process. It lists project perspective, success level (target), chosen metrics, trends and important comments. Since the period is 6 to 8 weeks, the measurements should be collected on weekly basics, and there should be at least one team meeting per week [10].

Table 2 is another project scoreboard; its scope is product. It also lists perspective, goal (success level), chosen metrics, trends and important comments. The measurements should be collected on weekly basics, and there should be at least one team meeting per week [10].

The use of scoreboard can enhance communications as it can keep team members, stakeholders, customers and leadership updated and informed on the project status and progress. How to deliver the scoreboard is also important, sharing the scoreboard during meetings and following up by emails is a common approach. The frequency of updating and distributing of scoreboard depends on the project complexity and duration. For 6-8 weeks period, weekly may be enough. Creating a well-designed, well-organized and consistent scoreboard can help keep the project healthy.

Table 1. A scoreboard covering 6-8 weeks (Scope: process)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Project Perspective** | **Target** | **Measurement** | **Trends** | **Comments** |
| Internal Developers | Technology Capability | production performance |  |  |
| Productivity | actual schedule vs. plan |  |  |
| Efficiency | time unit cost yield |  |  |
| Behavior Assessment | peer questionnaire and KPI |  |  |
| Communications | Minimum communication | number of messages |  |  |
| Maximum clarification of tasks | number of clarifying messages regarding tasks |  |  |
| Maximum team work | number of issues solved by messages from team members |  |  |
| Tasks | Setting achievable tasks | number of issues per task, number of tasks per day/week |  |  |
| Tasks finished within time and quality | number of tasks completed, number of software functionalities |  |  |
| Not excessive workload to team member | number of tasks per team member |  |  |
| Risks | Minimum | number of high, medium and low risks |  |  |
| Financial | Succeed | earned value |  |  |
| Prosper | market share |  |  |
| Budget | running cost, expenses |  |  |

Table 2. A scoreboard covering 6-8 weeks (Scope: product)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Project Perspective** | **Target** | **Measurement** | **Trends** | **Comments** |
| Customers | Customer Partnership | customer satisfaction |  |  |
| Product | percentage of purchase for the product |  |  |
| Project Delivery | on-time delivery and the ranking |  |  |
| Partial Results | Results according to project proposal | number of change requests |  |  |
|  | Result according to user requirements and expectations | number of change requests |  |  |
| Integrated Product | Has all modules integrated | percent of modules integrated |  |  |
| Has all functionality specified | percent of functions realized comparing to required number of functions |  |  |
| Has minimum bugs | number of test cases executed, number of bugs resolved |  |  |

**3)** **Contact software project developers/managers. Interview them regarding schedule slippage and rescheduling, present your finding.**

**Solution:**

The given problem is related to Chapter 14 in textbook [2]. First of all, the author emphasizes the importance of keeping schedule in project management, then the author gives steps to conduct analysis and make decisions when the schedule slipped. Then the steps to rescheduling are proposed and a case study is conducted.

I speak with Mr. Yao who is a senior developer at Bloomberg and Ms. Hu who is a developer and a project manager at J.P Morgan about the issues of schedule slippage and rescheduling during project management process, they basically agree with the procedures stated in the textbook, for example, when the schedule slips, developers/project manager should identify slipped tasks, quantify the amount of slippage, identify the dependent tasks, create multiple scenarios for schedule impact, finally reach a team decision on schedule slippage [2]. And for rescheduling there are two approaches: forward scheduling and backward scheduling. They shared their industrial experience in project management, and what an observant project manager should do when schedule slips happen in different stage of project.

1. Early stage schedule slips

During this stage, many can be changed, like to clarify project requirements, to change delivery date and milestones, to increase project cost and budget, add more resources, or change working environment.

1. Midterm project schedule slips

During this stage, the project may in detailed design or early implementation. If schedule slips happen, the easiest way is to slip the delivery date, increase project budget, or change the way resources work. The implementation may not reach its milestone, but with reviews and inspections it may reduce rework and testing time.

1. Late stage schedule slips

It is very difficult to recover schedule slips during this stage. The only option may be extending the schedule and increase project budge. Detecting the schedule slips as early as possible is essential for the project manager.

Other than the forward scheduling and backward scheduling approaches in textbook, they said the schedule slippage is common and rescheduling is quite often, they provide their own helpful tips to manage the risks of schedule slippage and share with me some useful links [5]. For example, as developers/project managers, we should create a well-managed schedule and reachable deadlines, we should take disciplinary action, acquire more resources and manage them effectively, enhance risk management, and we always maintain communications, offer incentive to team members, stakeholders, and third parties.

**4)** **Research, plan, and do a risk assessment on the deployment of a software system that includes a large databases. The new system replaces an existing system and therefore requires extracting existing data from the old database and loading this data into the new database.**

**Solution:**

All projects have risks must address from project launch through product delivery. The Chapter 5 in textbook lists steps to assess risk as following [2]:

1. List the risks.
2. Rank the risks by likelihood.
3. Rank the risks by impact.
4. Combine the risk likelihood and risk impact lists.

Data migration is the process of transferring data from one system to another system using a variety of tools and techniques. By doing literature review [6] [7] [8] I can list some most common risks of data migration:

* Data Loss – some of data may not migrate over from the source system.
* Data Inconsistency – data quality and completeness may suffer; semantics errors may occur.
* Data Corruption – unwanted data can be migrated into the new system.
* Extended Downtime – when data migration process takes longer than expected.
* Application Stability – the target system can be unstable for a number of reasons.
* Orchestration – the processes of data migration are not performed in order. The order of data migration is extremely important as there are varied dependencies between various objects.
* Interference – multiple users make use of the application simultaneously during the migration process.

Following the above step 2, 3, and 4, I list risks in the table ordered by combined rankings (based on Likelihood rank and Impact rank, the most likely risk is ranked 1, the risk with the largest impact is ranked 1):

Table 3. Risks of Data Migration Ordered by Combined Ranking

|  |  |  |  |
| --- | --- | --- | --- |
| **Likelihood Rank** | **Impact Rank** | **Combined Rank** | **Risk Description** |
| 1 | 4 | 5 | Data Inconsistency – data quality and completeness may suffer; semantics errors may occur |
| 3 | 3 | 6 | Data Loss – some of data may not migrate over from the source system |
| 5 | 1 | 6 | Orchestration – the processes of data migration are not performed in order |
| 2 | 5 | 7 | Data Corruption – unwanted data can be migrated into the new system |
| 6 | 2 | 8 | Application Stability – the target system can be unstable for a number of reasons |
| 4 | 7 | 11 | Extended Downtime – when data migration process takes longer than expected |
| 7 | 6 | 13 | Interference – multiple users make use of the application simultaneously during the migration process |

**5)** **Suggest changes to the software project plan described in our required textbook (i.e., *Software Project Management: A Real World Guide to Success, by Joel Henry*) that would make assessment easier and more effective.**

**Solution:**

The given problem is an improvement based on the content of software development plan in textbook [2]. The software development plan should begin from software project outline and should identify the software process the project will follow, the process should support project goals and attach project risks. The author suggests in order to write software development plan, it should cover all the important topics as following:

* Project Overview
* High-Level Functionality
* Project Staffing
* Software Process
* Schedule and Effort Estimates
* Measurements
* Risks
* Software Tools
* Hardware, Software, and Personnel Support

Although these points are detailed and elaborated and the process for project planning is solid, the author should consider tangible metrics that can be tracked over time into software project planning [9]. For example, in software project process part, we could track metrics like the percentage reduction in rework or retest. In project risks part, we could track risks by rankings.

Another aspect the author should emphasize more is the flexibility. Software project plan must be flexible enough to adapt to the unforeseen changes that always occur within dynamic environments and can be tailored to the changing requirements.

The last aspect I suggest the author to consider is to make cross-project plans that cross multiple projects within the development company, as most of companies have multiple projects operating simultaneously, the project manager should ensure utilizing resources effectively across these projects.

**References:**

1. Getting Started (Compact 2013) <https://docs.microsoft.com/en-us/previous-versions/windows/embedded/jj200349(v%3dwinembedded.80)>
2. Joel Henry, Software Project Management: A Real-World Guide to Success, ISBN 0-201-75865-2.
3. Elizabeth Harrin, How to Build A Project Scorecard, <https://www.projectmanagement.com/blog-post/26027/How-To-Build-A-Project-Scorecard>
4. Lindsay Scott, What is a Project Scorecard? September 26, 2017, <https://www.arraspeople.co.uk/camel-blog/project-management/what-is-a-project-scorecard/>
5. Michael Guerrero, Helpful Tips to Manage Project Slippage, <https://www.brighthubpm.com/monitoring-projects/123098-ten-tips-for-keeping-your-project-on-track/>
6. Potential Risks in Data Migration Process, <https://icedq.com/data-migration/the-data-migration-process-and-the-potential-risks>
7. Kate Miller, 7 Reasons Data Migrations Fail, <https://www.premier-international.com/articles/7-reasons-data-migrations-fail>
8. Tasmanian Archive + Heritage Office, Information Management Advice 60 Part 5 Successfully Manage Information Risks During System Migration, <https://www.informationstrategy.tas.gov.au/Records-Management-Principles/Document%20Library%20%20Tools/Advice%2060%20Introduction%20to%20Risk%20Management%20Processes%20Part%205%20-%20Manage%20Information%20Risks%20during%20System%20Migration.pdf>
9. Rohit Sharma, 10 Ways to Improve Software Project Planning, February 21, 2017, <https://dzone.com/articles/10-ways-to-improve-the-software-project-planning-1>
10. Ljubica Kazi, Narendra Chotaliya, Balanced Scorecard Framework in Software Project Monitoring, July 2011, ResearchGate, <https://www.researchgate.net/publication/267943773>