

The following is a list of lessons learned compiled from various IT development projects. The aim of providing Lessons Learned on the project Final Report is to provide Project Managers with a record of what worked and what did not work. Please note that this is not a list of technical issues, rather, they are recommendations on Project Management.

Change Management

- Ensure that expectations are clearly articulated and affected employees have “buy-in” to the directional change. Allow appropriate time for employees to see the benefits of the change, so that eventually they will accept and support it.

Communication

- All Stakeholders, including the Vendor and other external organizations should be part of the Communication Plan.
- Communications should be continuous throughout the project. Engage CSUs and Service Desk more frequently. Do more demos to key support staff.
- Important to have meetings between internal and external team members. These types of sessions should be mandatory for new technologies that we introduce.
- Provide documentation on product to teams very early on, to ensure that incompatibilities are identified at earlier points in the project.
- Provide clearer instructions to Functional Users regarding timeframes. Improve monitoring of Functional Users’ activities or progress. Make this a part of the monthly Status Meetings.
- Separate meetings should be set up among technical staff for design and architecture; and these should occur earlier in the project. Campus-wide emails (allstaff) should be used in the execution of the Communication Plan.
- Make communication issues to superiors a formal task.

Project Management Planning

- The CCS Information Security group should be part of the early planning stages of any project.
- Detailed Minutes of meetings should be kept for the recollection of past meeting discussions.
- Establish a clear Disaster Recovery vision.
- Do extensive up front planning, but be prepared to modify your approach, timelines and budget as new unanticipated requirements surface.
- Clearly define Milestones. Projects with too many Milestones should be broken up into smaller projects.
- Large projects should be reviewed with the users to determine if a phased approach can be implemented. Most users may not be aware of how this approach would work so educating them may open up the possibility of building in phases.
- Arrange an orientation session prior to starting the project. Have fewer inexperienced resources, more specialists.
- Training was not geared to the users in the audience; should have spent more time planning the training with the vendor to make better decisions on focus, timing, and audience.

Requirements Planning

- Perform walkthroughs of Requirements, Technical Architecture, etc.
- Obtain all relevant documentation from vendor.
- Conduct systems architecture analysis during Requirements phase.
- Dedicate more time to the business requirements gathering phase. More time is needed for a Business Analyst to analyze the details of the existing process and meet with various team members. Smaller requirements gathering sessions also recommended.
- Extra time in planning to conduct thorough architecture and design reviews in order to flush out all the design considerations early. Ensure that stakeholders have reviewed the requirements /decisions made prior to development.
- For any infrastructure project, develop a System Design/Technical Requirements document identifying architecture, environmental, IPs, ports, disk space & file allocation, software versions/licensed required. This would reduce some of the technical issues encountered and reduce time spent on setup and configuration.
- Mapping and documenting business processes in the planning stage would have facilitated the gap analysis process and provided a basis for process improvements in light of new systems.
- Policy issues should be clarified and identified in the Requirements.
- Third party software integration should always be reviewed thoroughly as part of the initial project planning and scope (part of Requirements & Planning). Early investigation allows CCS to determine technologies being used. Investigate integration strategies and discuss flexibility with the 3rd party.

Resource Planning

- Key technical resources need to get together at the outset and agree on a unified approach, rather than having one group do a lot of work only to have the effort precluded by a different approach. Planning should include all members of project team.
- Be aware of time of year and have resources provide dates they are not available (on annual leave), so that the Schedule could be adjusted accordingly.
- Training requirements should be included as part of the Project Plan so that time and money are allocated accordingly.
- Put some thought at the beginning of the project into the selection *criteria* for pilot participants. Should all faculties/departments be represented? If not, which? Should participants with no previous experience with the product be included, etc?
- Time and resource estimates from previous upgrades should be used as comparable projects for estimation. For subsequent upgrades, keep the role of Project Manager and Testing Coordinator as separate resources.
- Projects with a large scope and lengthy timeline should always have a dedicated PM, BA and Lead Developer in addition to the SME. This minimum of 4 members will ensure better communication and project organization.

Technology

- For projects where the technology is new to Carleton or when a project has many unknowns, allow extra time in the Schedule for unanticipated events. Run a small pilot project when testing new technology.
- Don't be afraid to change systems or technology even after a good chunk of the work was completed. When the application change was made, the project went from being very late to ahead of schedule.
- Conduct systems architecture analysis beforehand to ensure integration capability with existing

applications.

Testing

- Test Plan should be communicated to all resources on the project. Ensure that time is spent before testing commences so that developers and testers are in synch on the use of test scripts, PRs, reporting processes, enhancement lists, etc.
- Standardize test script format for use across projects. When new functionality is introduced, test scripts used in project should be appended to master list for use in future upgrades.
- Set clear consistent standards for all testing documents; e.g., test methodology, for all groups. Create or provide a clear high level test plan example for all groups. Create or provide a clear low level test plan example for all groups.
- Ad hoc testing created a lot of “back and forth” activity and increased the volume of testing and re-testing. Develop actual test cases before the project goes forth.
- Expand testing of web applications to include all browsers: IE, Firefox, Safari, Google Chrome, Opera, etc.

Vendors/External Services

- It is important to consider the Vendor’s customer service levels before contracting services with them.
- During complex service transitions, it is valuable to have the professional services of the Vendor available on site.
 - Must know exactly what we want before Vendor shows up to implement; i.e., Statement of Work.
 - Avoid engaging with companies that have limited resources; i.e., owner is also the consultant, designer and SME.