In 2002 a survey was conducted to assess current requirements engineering techniques usage in industry (Neill and Laplante 2003). The survey revealed industry trends regarding the specific practices used by individuals in their most recent project. Key findings of the study included:

- Formal models were rarely used.
- Ad hoc development practices did not impact end-product quality.
- The waterfall lifecycle model was still popular.
- Object-oriented techniques were not dominant.

Respondents also indicated a significantly higher rate of software project success than is often cited by industry surveys. In addition, the survey revealed that project size and duration were better indicators of success or failure then were lack of use of defined requirements engineering techniques.

Have things changes significantly in the six years since this survey was conducted? Have new trends emerged? We launched a new survey to try to answer these questions and some of the results are presented here.

**Survey design and conduct**

The new survey instrument largely replicated the 2002 survey (Table 1). Each of 22 questions of the original survey were represented along with five new questions. Unlike the 2002 survey questions were grouped into seven categories of study [Marinelli].
Table 1: Summary of Survey Questions. Key: D -- dichotomous, LS -- Likert Scale, MC1 -- multiple choice (single selection), MCN -- multiple choice (multiple selection), OE -- open-ended, SDS -- semantic differential scale.

In order to remain consistent with the target population sampled in the 2002, the new survey was targeted to prospective, current, and past graduate students in the Masters of Software
Engineering and Master of Science in Information Sciences programs at the Penn State Great Valley School of Graduate and Professional Studies (PSGV). For prospective students, the pool was limited to applicants who had applied in the two years prior to the start of the survey. PSGV serves mostly adult, working professionals in the Greater Philadelphia Metropolitan area. Therefore, survey participants were derived from well-known companies in the region such as Lockheed Martin, QVC, Unisys, Vanguard, Boeing, Siemens, Verizon, and from numerous, smaller, local companies.

The survey was implemented using the web-based QuestionPro survey tool (www.QuestionPro.com). Recruitment emails and reminders were sent to prospective students between April and June 2008.

Survey results

The enrolment profile for the survey indicates that of the 1855 subjects that received an invitation to participate in the survey, 93, or 5%, interacted (where interaction is defined as data entry into one or more fields). This response rate is significantly lower than that for the 2002 survey (12%). We are not certain why the response rate was lower. Some variation could be due to increased employee mobility since the first survey. Another explanation could be the use of a different survey tool. In any case, we do not consider the difference in response rates as a major threat to validity.

Respondent Characteristics

Survey respondents described themselves as programmers, developers and software engineers 52.08% of the time. 31.25% of respondents characterized themselves as architects, project managers or systems engineers; positions typically involved in the higher-level aspects of computerized system’s technical design (Figure 1).

![Figure 1: Respondent's position (Question 21).](image)

The respondents in this survey more frequently identified themselves as developers than project managers than in the 2002 survey. Survey results again indicated that respondents worked in a wide range of business domains (Figure 2). This distribution was very similar to that of the 2002
As in the 2002 survey, respondents had experienced a varying number of projects in the last five years from three or less to more than 26. The mean experience level in the 2008 survey (question 22) was somewhat lower (10.15) than the that of the 2002 survey (12.5). However, in both surveys, the distribution was positively skewed with the median of the 2008 survey being 7 while that of the 2002 survey being 5. This indicates that the 2008 population was made up of a more tightly grouped set of respondents that were marginally more experienced than those in the 2002 survey. Respondents of the two surveys came from very similar work environments both in terms of number of staff and annual budget (questions 24 and 27).

**Project Characteristics**

Respondents were asked to base their project responses on one project that they were either currently involved with or had taken part in during the past five years. Reported projects were distributed across a broad range of application domains, with a bias towards applications in the financial sector (Figure 3).
Given that the sample populations were similar, it is no surprise that the distribution of project domains was consistent between surveys.

Most projects (60.00%) took a year or less to complete with 16.67% taking between 24 and 48 months, and only 10% taking more than 48 months to complete (question 3). In 2002 nearly half of the projects were listed as of one to two year duration. This change may be an indication that delivery times have been reduced in the intervening years between the surveys. Project size, in terms of the number of IT staff assigned (question 4) was largely unchanged between the two survey years.

The 2008 survey also looked at project size in terms of lines of code. The projects experienced were predominantly of small to medium size in terms of lines of code. 70.37% of projects contained 50,000 lines of code or less. This data was not collected in the 2002 survey.

**Software Development Practices**

59.26% of respondents either did not use a Software Quality Management (SQM) approach or were not aware of the approach applied. ISO9001, Six Sigma and CMM/CMMI SQM approaches were used in only 22.22% of projects (Figure 4).
Figure 4: Software quality management approaches (Question 6).

68.52% of respondents either did not use a Software Development Framework (SDF) or were not aware of the framework applied (Figure 5).

Figure 5: Software development frameworks used (Question 7).

Other responses included “Agile”, “Waterfall”, “.NET”, “PowerBuilder”, “third party software”, and “Home Grown Company Process”. This question was not asked in the 2002 survey.

65.38% of respondents reported using no Application Lifecycle Management (ALM) software or were not aware of the ALM application used. This low rate of adoption is not surprising given that these tools have a high process overhead and lend themselves to large projects and teams. A total of about 10% reported using either Visual Studio, IBM Rationale Software Suite, or “other.” Other responses included “Home Grown” and “PowerBuilder.” This question was not asked in the 2002 survey.

In contrast to the high number of respondents that responded “None” or “Unknown” to the use of SQM, SDF, and ALM systems, nearly all respondents were aware of the Software Development
The inclusion of agile methodologies and the expansion of the number of SDLC methodologies offered as options in the 2008 survey make some of the other results for this question difficult to correlate. However, given that the respondent, business, and project profiles are very similar between the two surveys, it is quite surprising that the results of the SDLC methodology are so similar. In the 2002 survey, 35% of respondents reported using the Waterfall SDLC methodology. The 2008 survey found that 33.33% used Waterfall, suggesting that little change has taken place in the degree of adoption (or abandonment) of the Waterfall SDLC methodology. A positive note is that a significantly higher proportion of respondents reported using prototyping. Nearly 75% of respondents in the 2008 survey reported using prototyping while 60% reported using prototyping in the 2002 survey.

Three quarters of respondents indicated that prototyping was employed in their project. Those 39 respondents that used prototyping used an average of 1.3 prototyping methods per project. Evolutionary prototyping was selected by 24 respondents, making it the most popular method selected (Figure 7). The increase in evolutionary prototyping over UI prototyping may be indicative of an increase in the use of evolutionary and agile methods since the earlier survey.

Figure 6: Software development lifecycle employed (Question 9).
Interestingly, six of the respondents that reported using evolutionary prototyping also reported using Waterfall, V-Model, or Model-Driven Development as their SDLC methodology.

Respondents reported using, on average, three to four requirements gathering techniques, with the range being between one and seven. The most popular technique reported was “Interviews”, with “Use Cases” and “Storyboarding / Whiteboarding” being the second and third most popular. Of interest is that Use Cases, which are most often associated with the Unified Process, were reportedly used by seven respondents who indicated that their project followed Waterfall SDLC methodology. Similarly, eight respondents that reported using Storyboarding indicated that their project followed Waterfall SDLC methodology (Figure 8).

There were some interesting shifts in the profile of Requirements Gathering techniques employed. In the 2002 survey, Use Cases and Scenarios were the most popular methods employed. In the 2008 survey, these categories were split into two; however, adding them together, they are still the most popular at ~26.5% of the total frequency. Storyboarding and
Marinelli and Laplante Survey

Interviews have emerged as popular methods with Interviews being the most frequently selected option in the 2008 survey.

Only 22 of the 49 respondents that answered the question regarding Requirements Analysis and Modeling techniques reported using any methodology. Of those 22 respondents, five reported using two techniques while the remaining 17 respondents reported using one technique. The most popular techniques employed were Object Oriented Analysis and SSADM at 16.98% (Figure 9).

In the 2008 survey, significantly fewer respondents reported using no Requirements Analysis and Modeling techniques than in 2002. Of those that did report performing analysis and modeling, the distribution of techniques used is not significantly changed from the earlier survey. Similarly, no change has been observed in the use of Requirement Specification Notation systems or Requirements Inspection Techniques. The distribution of results for these two questions is nearly identical to the 2002 results.

The majority of users (53.19%) still report that requirements are expressed in terms of natural language (Figure 10).
It is discouraging that only 30% of users report utilizing semi-formal notations such as UML and that formal methods are still infrequently used. It is also of interest that formal specifications techniques are still not commonly utilized (6.38%). Also of interest is that there is no discernable relationship between specification notation chosen and SDLC methodology utilized.

Twenty six of forty nine respondents (53.06%) reported that they performed requirements inspections. That group of twenty six respondents used, on average, 1.6 techniques. Formal walk-throughs, ad-hoc walk-throughs, checklists and scenarios were all quite popular (Figure 11).

No respondents reported using automatic requirements inspection tools – surprising in light of the fact that two respondents reported formal specification methods for requirement specification.

**Software Quality and Productivity**
Respondents were asked a series of questions meant to assess the level of quality and productivity the project achieved. As with the 2002 survey report, we forgo a question-by-question discussion and provide summary results instead.

Overall, respondents agreed that the quality of their software was good and that it met end user needs (strongly agreed 68.75% of the time). However, questions relating to delivery timeline indicate that the projects represented in this study took longer than the respondents had expected to deliver 55.83% of the time (strongly disagreed or agreed with the statement that “project duration was on schedule”).

With respect to their level of satisfaction with regards to the SQM, SDLC, and RE methods applied in their project respondents reported that they were either Satisfied or Neutral in their level of satisfaction with these efforts 57.44%, 68.75% and 68.75% of the time, respectively. Of interest is that no single SDLC or SQM approach emerged as significantly more favorable than any other. However, MDD and Chaos SDLC approaches were rated significantly lower than other categories. Data for this question are reported in Table 3. When the same questions are asked with regard to the respondent’s company, the results are similar. However, a large number of users report being dissatisfied or extremely dissatisfied with the SQM, SDLC, and RE approaches employed at their company (47.50%, 37.50% and 32.61%, respectively).

When respondents were asked if they felt that sufficient SQM, SDLC, and RE efforts are being applied at their company, the responses were evenly divided. Only 47.92% of respondents felt that their companies were doing enough (agree or strongly disagree) to manage quality, perform adequate requirements engineering, and to adhere to a specific SDLC discipline (question 17). These results are surprising in that the companies represented in this survey are generally large.

Comparing the measures of software quality between the 2002 and 2008 surveys yields interesting results. It appears that product capability and ease of use has improved since the previous survey with significantly more respondents agreeing with statements to that effect. However, delivery delays seem to be a bigger problem in the 2008 survey than they were in the 2002 survey. This is interesting in light of the fact that the project durations reported were considerably lower in the 2008 survey versus 2002. Respondents of the 2008 survey were considerably happier overall with the abilities as well as the work product of their development teams. This could be due to the change in the survey population. The 2002 survey group consisted largely of project managers and systems analysts, whereas the 2008 group consisted mainly of developers and software engineers.

In this version of the survey, respondents are evenly split over the question of whether or not their company performs an adequate amount of requirements engineering. This appears to indicate no significant improvement over the 2002 survey results.

**Analysis of Software Quality Measures**

Response data was analyzed to determine if measurable differences in the respondents’ satisfaction with the product and with the RE practices applied on the project could be correlated with responses to questions related to Software Development Practices (Questions 6-10). We
note a few of the findings that are provocative and may signify emerging trends.

For example, we found that when usage of Software Quality Management tools is compared to respondent feedback regarding quality and productivity, the data indicates that users of SQM tools realized modestly better productivity than users that either used no tool or were unaware if a tool was used. More interesting is that respondents in the positive group (Used SQM) provided higher satisfaction scores in the Subjective Analysis sections. Note that results were combined into SQM users and those users that responded “None” or “Unknown”. This is a profound result in that it may indicate that the commitment to quality and process dictated by the adoption of an SQM framework has a positive effect on the outcome of a project.

When examined in the aggregate, users of Software Development Frameworks also provided answers that indicate an improvement in the quality and productivity when compared to users that either used no SDF or were unaware if one was used. In addition, users of SDFs generally provided more favorable answers to the Subjective Analysis questions than did those that answered “None” or “Unknown”.

An aggregate analysis of Application Lifecycle Management tool users indicated that there was no significant improvement in the quality and productivity when compared to users that either used no ALM tool or were unaware if one was used. ALM users generally provided the same level of satisfaction on the Subjective Analysis questions than did those that answered “None” or “Unknown”. Given the positive results observed for SQM and SDF responses it is curious that ALMs have not delivered on the promised productivity and quality gains. It may be that these technologies are still so new that they have yet to be fully leveraged.

Similar to the response to ALM use, the selection of a given SDLC methodology does not correlate well with quality and productivity measures, nor does the selection of methodology change the subjective assessments provided by the respondents. Even when the data was grouped and analyzed according to SDLC class (Sequential, Iterative, or Evolutionary), no significant differences were observed. This would seem to indicate that the actual method selected is less important to the team’s commitment to a given SQM methodology or SDF.

Conclusions

In comparison to the 2002 survey, a number of trends emerged.

- The Waterfall SDLC is still employed about one-third of the time.
- Use cases and scenarios are still the most popular requirements elicitation techniques.
- Semi-formal notations such as UML are used about 30% of the time and formal methods are still infrequently used.
- Storyboarding and interviews have emerged as popular requirements elicitation methods.
- Perception of product capability and ease of use has improved since the previous survey.
- Delivery delays seem to be a bigger problem in the 2008 survey than they were in the 2002 survey.

The present survey also provided a number of new findings:
• The commitment to quality and process dictated by the adoption of an SQM framework has a positive effect on the outcome of a project.
• There is no significant improvement in the quality and productivity when compared to users that either used no ALM tool or were unaware if one was used.
• The selection of a given SDLC methodology does not correlate well with quality and productivity measures, nor does the selection of methodology change the subjective assessments provided by the respondents.

Both the 2002 and 2008 survey had limitations. The survey respondents were of a similar background, geographical location, and education level. Thus, it is not possible to project these findings to a broader audience. Another consideration is that senior management and end users were not included in the survey. As a matter of perception, software development professionals tend to rate success achievement higher than these other groups.

With colleagues we are in the process of replicating this study in Malaysia and in the Near East to collect comparative data. We hope that this work will provide some interesting comparative data and we encourage others to replicate this survey with other populations.

References
