



The Quality Journal

A publication of the Chicago Quality Assurance Association
113 McHenry Road, Suite 218 Buffalo Grove, IL 60089 • www.cqaa.org

June 2011

In This Issue

Message from the President...1

CQAA News...2 to 4

Industry Perspective... 5 to 12

Another Look...13 to 17

Hot Topics...18

CQAA – The Organization...19

Event Calendar...19

Message from the President

We recently learned of some very exciting news for our chapter and those of us in the software quality industry. The QAI Global Institute has decided to bring the annual Quality Engineered and Software Testing (QUEST) Conference back to Chicagoland for 2012 to celebrate QUEST's fifth anniversary. The benefit for all of us is that we can enjoy a great event and wonderful insights from industry thought leaders in our own backyard. Our local chapter also benefits with opportunities to serve as the host of this internationally attended event and our members receive a 10% discount.

Over this summer, you will want to prepare in three ways – ponder the idea of developing and submitting a presentation to share with conference attendees; tell your colleagues and friends about the conference and, most importantly, secure funding in your annual budget cycle for yourselves and your team members to attend the QUEST Conference.

Check the web site, www.cqaa.org, for updates in the next few months or contact me directly if you wish to learn more about QUEST or suggest ideas on how our chapter can contribute to making QUEST 2012 the best yet.

In this issue, we discuss a few topics that seem to be major areas of interest for our industry. Thanks to Alan Cameron Wills of Microsoft who shares some actual experiences on working with the Agile project framework. Also, our own Cindy Glaser provides an insightful recap of the April Lunch & Learn discussing working with distributed teams. We hope you find both articles useful to dealing with your daily challenges.

Summer has arrived! We hope you find a few days to enjoy some excursions, picnics, concerts, or other great activities available to us in these next few months.

Nancy Kastl, President CQAA

CQAA
Thanks Our 2011
Sponsors

Diamond Sponsor
Microsoft

Lunch and Learn Series
SPR Companies



Save the Date

QUEST 2012

April 30 - May 4, 2012

Chicago is the hosting city for the 2012 North America QUEST Conference (Quality Engineered Software and Testing). QUEST will be at the Westin Lombard Hotel. Be sure to plan the QUEST Conference in your 2012 budget. Watch for more details. www.gaiquest.org



July Speaker Program

Mobile Technology Testing – Are You Ready?

Wednesday, July 27, 2011 – 1:30 to 4:00 PM

About the Topic

If the mobile technology train hasn't blown through your organization it soon will. Are you ready to jump onboard and face the unique testing challenges presented by mobile applications? Mobile visionary, Alex Bratton, and software quality and testing expert, Lee Barnes, will lead you on a journey to help you understand where mobile is, where it's going, and what you can do to help ensure mobile quality in your organization.

Alex will provide an overview of mobile technology and discuss why good process, from strategy through testing, is essential in ensuring the quality of mobile applications. Lee will highlight testing challenges specific to mobile apps, and present mobile testing best practices including techniques for leveraging test automation on mobile platforms. Attendees will walk away with a solid mobile technology baseline and best practices for addressing the challenges that lie ahead.

Key Learning Objectives

This talk will cover:

- State of the mobile industry and why it matters to you
- Why good process is even more important to ensure mobile application quality
- How to address the unique testing challenges presented by mobile

About the Speakers

Lee Barnes has over 17 years of experience in the software quality assurance and testing field. He has successfully implemented test automation and performance testing solutions in hundreds of environments across a wide array of industries. He is a recognized thought leader in his field and speaks regularly on related topics. As founder and CTO of Utopia Solutions, Lee is responsible for the firm's delivery of software quality solutions that include process improvement, performance management, and test automation.

Alex Bratton is a serial entrepreneur motivated by his passion for making technology accessible and easy-to-use. In addition to being the CEO for mobile-focused Lextech Global Services and Lextech Labs, Alex is an author, speaker and evangelist for the effective use of mobile apps in the enterprise.

Host and Location

JPMorgan Chase
Chase Auditorium
10 S Dearborn Street
Chicago, IL

Agenda

1:00-1:30 Sign-in & Networking (please do not arrive before 1:00 pm)
1:30-1:45 Welcome and CQAA Announcements
1:45-4:00 Panel Discussion (15 minute break at 2:45)



August Live Webinar **Testing @ Microsoft**

Tuesday, August 9, 2011 – 11:30 to 12:30 CST

About the Topic

How do you ensure that your product works across thousands of customer deployments around the world running in different configurations, languages, geographies and scale? Where to spend the QA effort when you know you can never complete it.

Learn the processes and techniques that Microsoft practices to deliver high quality world class products. Identify how to prioritize QA processes, testing types to automate, key automation infrastructure investments and mixing both traditional and agile/exploratory testing techniques that validates your product 360 degrees.

Discuss innovative methods to make manual testing effective using customer usage patterns, “Dogfooding” your products, Bug Bashes, unhappy path testing and provide an “Overall Goodness Factor” report of your product at any point in the development cycle. Also discuss key automation strategies such as – the right level of UI automation and when and how to write it, automation that measures and report key metrics to understand the performance characteristics of your application, investing in infrastructure that can simulate large scale environments.

Key Learning Objectives

In this talk we will:

- Learn the QA practices that Microsoft uses to deliver high quality products
- Learn when to automate and when not to automate your tests
- Learn writing effective UI automation.

About the Speaker

Vijay Vedantham has over 14 years’ experience, mostly in software quality management, more than 10 years of which at Microsoft Corporation. Extensive experience in testing of both client side and server side technologies such as natural language processing, real time collaboration, virtual machine management and test and lab management. Currently, as a senior test lead in the Visual Studio Test and Lab management team, responsible for delivering tools that are targeted for helping in both manual, automated and performance testing of software applications.



CQAA News (cont.)

CQAA Sponsor Program Highlights

CQAA offers a number of sponsorship opportunities which are outlined below. Sponsorships help fund a variety of events and activities and serve to minimize participation costs for CQAA members.

CQAA Annual Sponsorships

Service Provider Sponsorship

Visibility through your company logo, description, and website URL incorporated on the CQAA website as a featured sponsor and recognition of your company as a CQAA sponsor in a variety of ways including recognition at events and CQAA media resources.

Enterprise Sponsorship

Recognition of your company as a CQAA sponsor in a variety of ways including recognition at events and CQAA media resources along with discounts for all employees at a variety of QAI and CQAA events

CQAA Dinner Events Sponsorships

CQAA has two dinner events each year, one in the spring and one in the fall. The dinner event features an industry renowned keynote speaker. Sponsorships include **Speaker**, **Information Table**, and **Dinner Table**.

CQAA Contributors Program

Donations of funds of any amount are accepted as well the opportunity to provide a facility and refreshments for CQAA program events.

Certification Highlights



Professional certification does make a difference for many organizations. In addition to the actual accomplishment, those achieving this status gain valuable knowledge in the profession. The QAI Global Institute provides the certification and local chapters provide some level of support to prepare for certification.

CQAA actively supports and hosts study groups and Prep Courses in the Chicago area for the professional certifications listed below. **Study Groups** are provided at no cost to the attendee and are formed based upon the number of people interested and typically meet weekly for a period of several months. **Prep Courses** are delivered over a two day period. Location and cost of Prep Courses is shown below. Go to www.cqaa.org/certifications for more information and to register.

CQAA Job Posting Board

Are you looking for some great people to work with? Whether you are looking to add to your team or looking to join a team, you will want to explore the CQAA Job Posting Board. Since its launch, we are seeing increased use by companies looking to find software quality professionals. The cost is quite reasonable for companies to post their available positions and it makes perfect sense to connect with the CQAA to find the right talent. If you are in the market, keep checking as new postings appear on a regular basis. The link to the service is on the main page of the CQAA web site.



Industry Perspective

Industry Perspectives are offered by recognized professionals for the purpose of sharing information with our members on approaches, techniques, or information that may be useful to professional growth and development. The content is the property of the author and any reproduction of this content outside of this publication is not permitted. Anyone interested in this content or the information conveyed in this article should contact the author directly.

About the author...

Alan Cameron Wills is a Senior Programming Writer at Microsoft, where he has also worked as a developer on modeling tools and process guidance. From 1994-2004, he led an independent consultancy specializing in modeling and requirements gathering.

His company's website is www.microsoft.com E-mail him at awills@microsoft.com

Using Models with Storyboards in an Agile Process

Alan Cameron Wills, Microsoft Corporation, © 2011 Microsoft Corporation

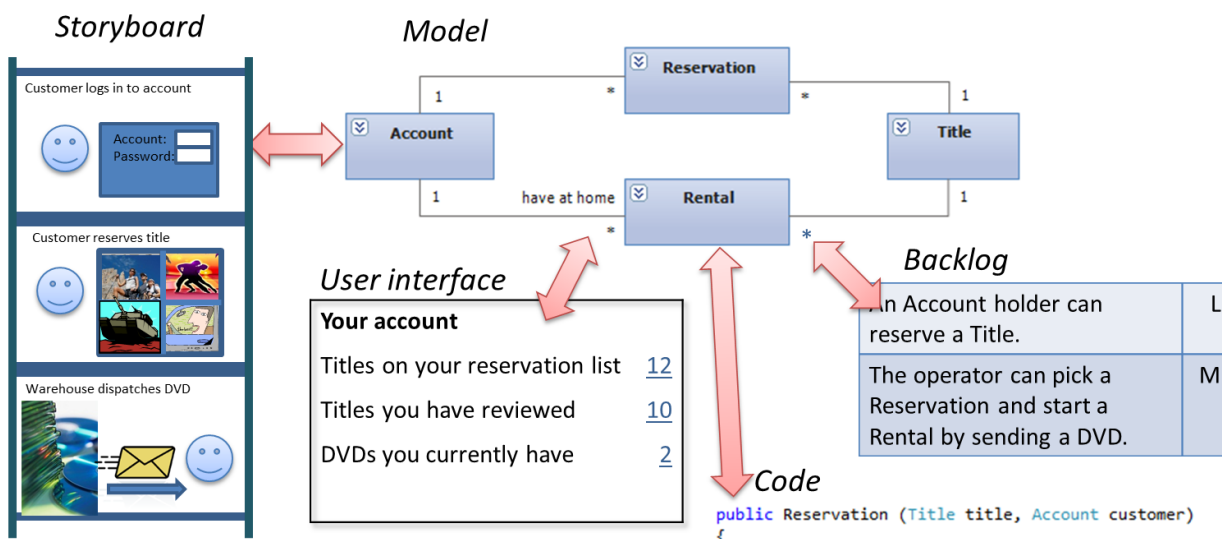
Understanding what your users actually need is notoriously the most error-prone part of software development, and the most expensive to fix. A good test regime can fix bugs in the code, but you can still deliver a bug-free application that doesn't fit the business. Even if your customer were to state the requirements unambiguously and without gaps or inconsistencies (!), the world around the users moves continually on, and the arrival of your application in itself changes how they work. Agile development methods therefore focus on early and frequent demonstration of tested working code, so that customers can review progress and call for course corrections as necessary. Correspondingly, modern development teams (outside safety-critical and embedded areas) rightly de-emphasize any need to tie down the requirements in fine detail at the start of the project.

Nevertheless, there is a balance to be struck somewhere between obsessive specification and not discussing the customer's needs at all. Agile teams create a product backlog – a list of briefly-stated user stories – from which the project is planned; but there is a variety of techniques for getting the background information from which the backlog items are written. Each item states what the users will be able to do, once that piece of development work has been completed. If the team is very familiar with the domain, that might be sufficient. But if they are, for example, independent software vendors entering a new area, then there is work to be done to acquire a good understanding of the client's business.

Where the team is large and/or dispersed, it is important to have other documents that help to explain the vocabulary of the backlog. Most of us have seen bugs that can be traced back to different understandings of the meanings of terms.

Industry Perspective (cont.)

This article is about two techniques that help you in discussions about the users' needs. Many people are familiar with *storyboards* – cartoon strips that envisage how the users will work with your system when it is complete. Less widely used, but in my experience very useful, are *domain or requirements models* – descriptions using part of UML in which you can define the “ubiquitous vocabulary” of the project: the terms that are used by customers, developers, in the user interface, in code, and in planning. These models can play a vital role in helping to ensure that what is written on each backlog item should convey the same meaning to both client and developer.



Storyboards and models can work well together. My emphasis here is on using these tools not just for communication, but also as active aids during discussions with your customer. By validating one view against another, they can help you ensure both that you have covered the important ground, and that you are using terms consistently.

The key to getting real value out of a model is not just to sketch it and stand back to admire it: make it do some work!

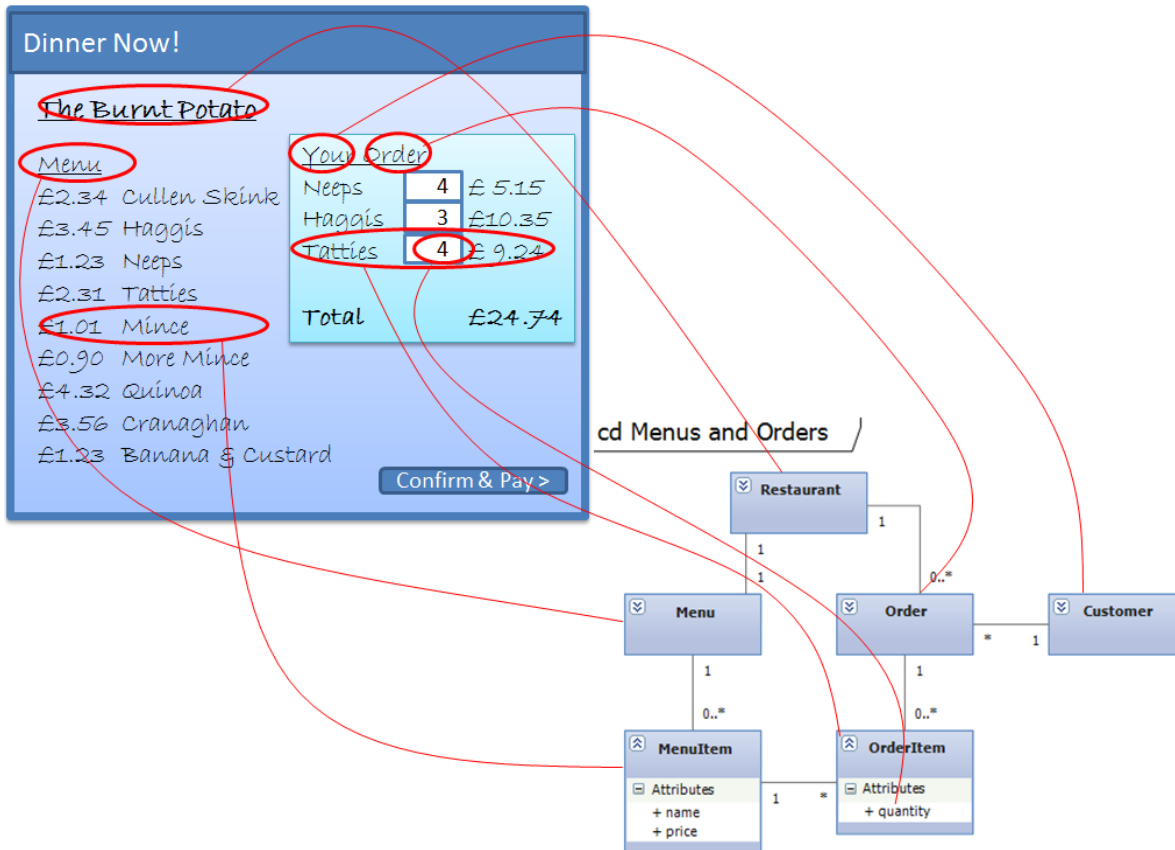
Domain Types

A model in UML can include several different styles of diagram, but in this article I'll focus on class diagrams; and in particular, class diagrams used specifically for modeling domain types.

The ‘domain’ in ‘domain model’ means that we’re not discussing classes in the code, but types of entities in the world that is visible to the user, either because they are visible through the user interface, or because they’re actual real-world entities. The purpose of the domain class diagram is to describe clearly the entities and relationships that the users and the software have to talk to each other about. The internal design of the software is excluded from a domain model, because we want to use it as the common basis for discussions with clients, and for the application’s dialog with users.

Industry Perspective (cont.)

Identify types by inspecting your storyboard slides. Storyboards typically include screenshots such as this one, in which we can find a number of types and relationships:



Draw relationships between the classes. Think of them as representing data, rather than message paths. Ask questions about the cardinality of each end of the associations: How many Menu Items can an Order Item refer to? How many Menus can one Restaurant have?

Your client is the authority on these questions, which often lead on to deeper discussions about less obvious assumptions. The model helps you make sure that you have the answers.

You don't need operations on the classes in a domain model. Dynamic aspects of the domain are represented in your storyboards. Assigning operations to classes is part of your software design, which is not what we're engaged in at this point.

Looking at each storyboard slide, you will typically find a different group of relationships. Put these in different class diagrams. Some of the same classes will appear in both diagrams. For example, a second diagram might include the type 'Customer Recommendation', which would be a new type, but associated to Restaurant and Customer which we've encountered already.

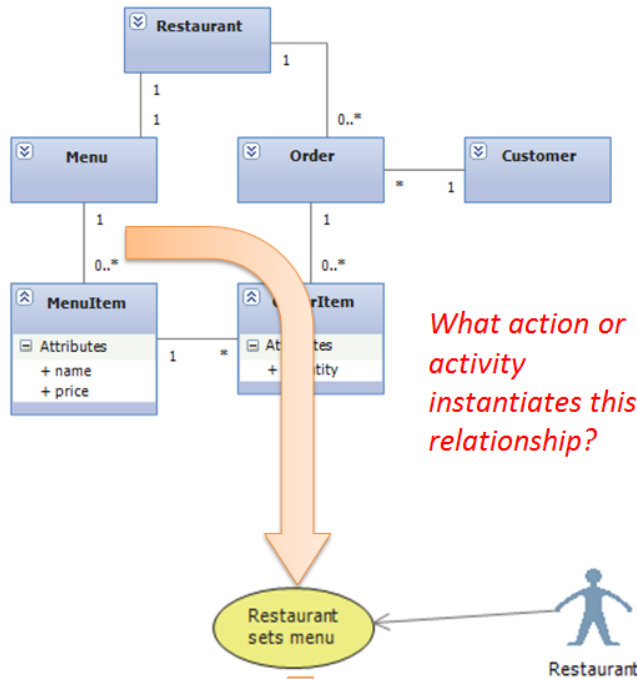
Industry Perspective (cont.)

Find more activities

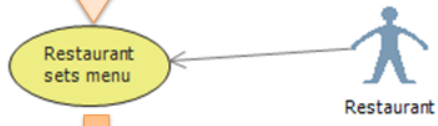
Now we can start putting the model to work to verify that we've covered every aspect of the system. Look at each relationship and ask: What actions create instances of this association? What actions delete it? Look at each attribute and ask: What actions change this value?

Make sure that your storyboards answer these questions. For example, what adds the Menu Items on a Menu? Again, this is a basis for discussion with your client. Perhaps there is a separate web interface for restaurant managers:

cd Menus and Orders /



What action or activity instantiates this relationship?



Provide storyboard detail



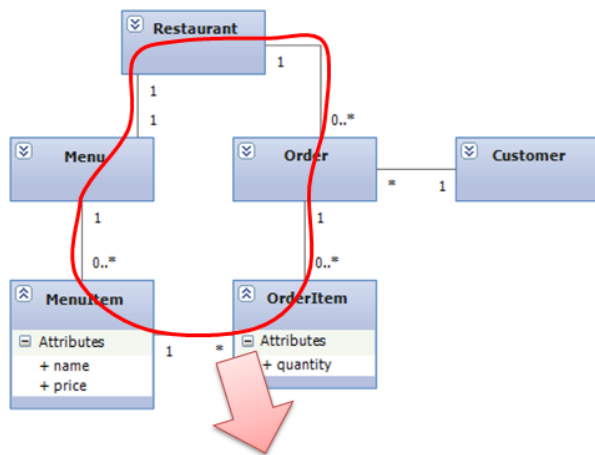
Industry Perspective (cont.)

Find business rules

Look for loops in the relationships in class diagrams. Ask whether every configuration of entity instances allowed by the diagram would actually be valid for the business. Do the instances have to form a loop, or must they never form a loop, or is it optional? It is sometimes helpful to draw an instance diagram as an experiment.

For example:

cd Menus and Orders



The Fearful Haggis:
Restaurant

Menu

Boiled Chips:
Menu Item

The Tartan Neep:
Restaurant

Order

Order Item
Quantity = 5

Question:
Is this allowed?

Answer: No



Business rule:

An order for one restaurant can't include menu items from another.

Consider also what combinations of attribute values are valid or invalid.

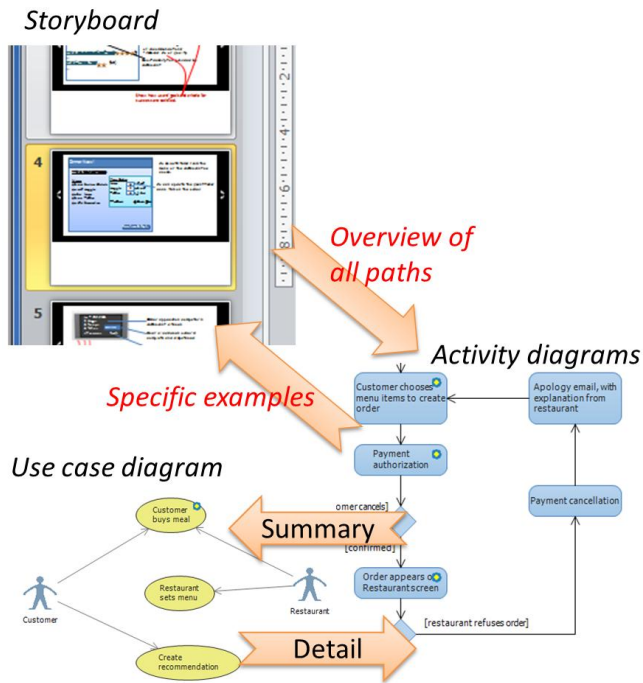
The point about this technique is that it throws up a question. Your client is the authority on the answer. Asking these questions will typically lead on to clarify other related issues. Therefore, to get the best value from creating these models, draw them while the clients are present.

Activity and Use Case diagrams

A plain slide presentation just lets you tell a linear story. But if you draw your storyboards on a whiteboard, you pretty soon do the obvious thing and draw branches and loops, using arrows between the story's frames – thereby turning it into an activity diagram.

Industry Perspective (cont.)

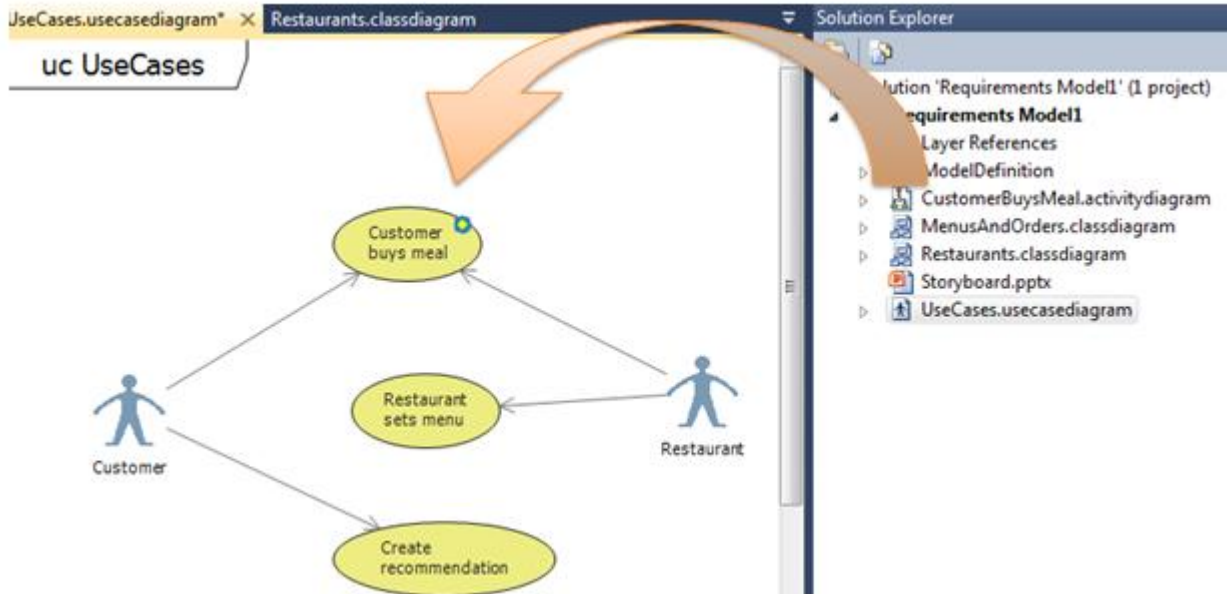
Back on the computer, if you don't yet have a neat tool that lets you draw those arrows between the slides, then you can at least draw a UML activity diagram. Use the storyboards to tell linear stories, and the activity diagrams to show all the alternative paths. If your tools allow it, create links between the corresponding slides and actions:



(If you are a Visual Studio user, there is an extension you can get that lets you link PowerPoint slides to any UML element.)

Each storyboard and/or activity diagram shows how a particular type of user achieves a specific goal – the customer buying a meal, the restaurant setting the menu, and so on. It can be useful to summarize these activities and users in a use-case diagram. Each use-case corresponds to one whole storyboard or activity diagram, and if your tools let you link them up, then so much the better:

Industry Perspective (cont.)

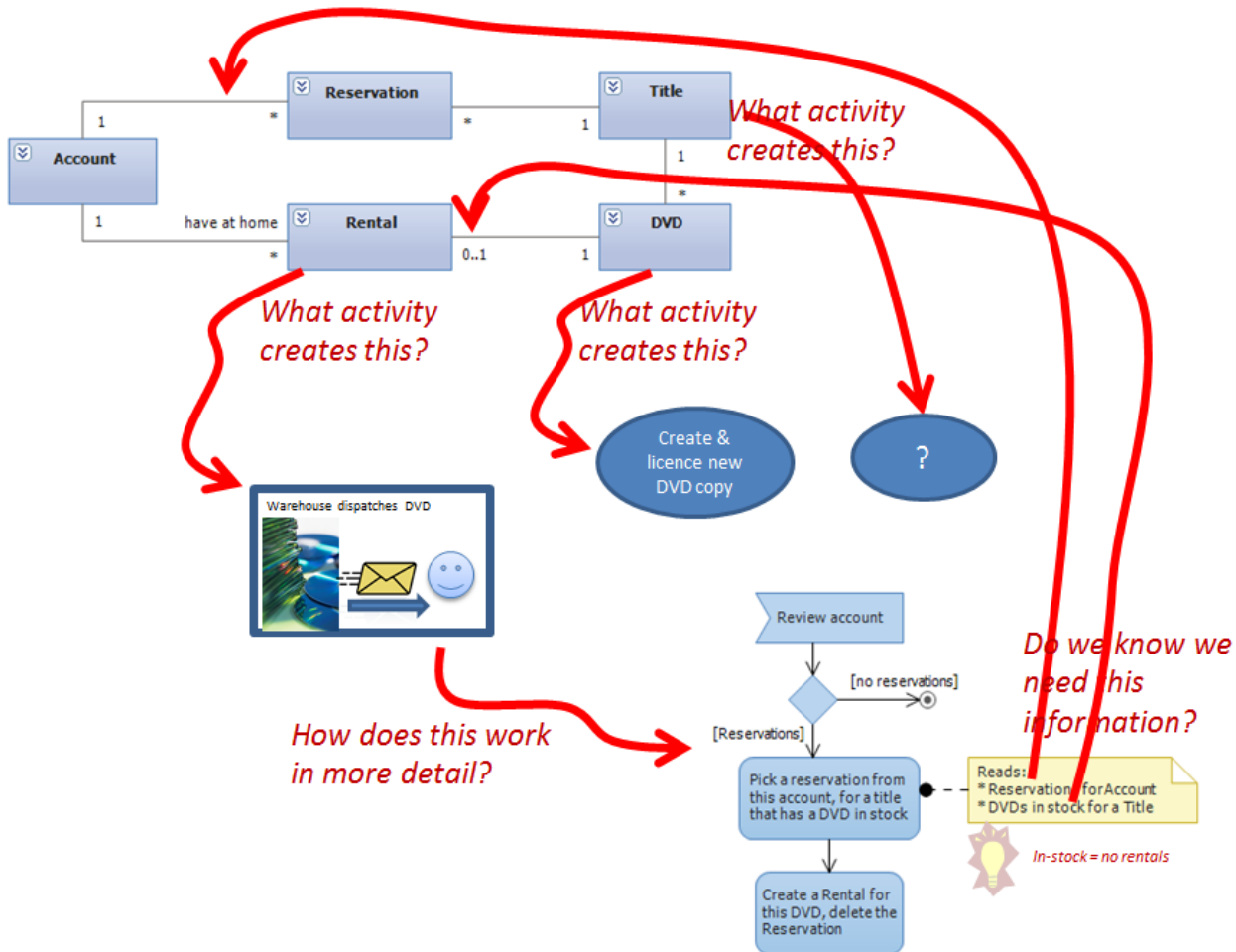


Cycle until consistent

As you discuss the business domain with your clients and other stakeholders:

- Whenever you create new types, attributes or associations, think about what actions create, delete, or update them.
- Whenever you create new actions - whether in the storyboard or on diagrams, make sure the data that they use or update are represented in the model.

Industry Perspective (cont.)



Following this cycle raises questions that you might not otherwise have uncovered until much later in the project.

When to use models

Storyboarding and modeling are activities chiefly for the early days of a project. You'll often revisit them, particularly in the first day or two of a sprint, when you will adjust them in the light of customer feedback, and add more detail about the upcoming work. But demonstrating working code is the ultimate and best way to test the correctness of your understanding of what is required. Avoid any obsessive pursuit of the fine details while you could be getting on with writing code. Depending on the familiarity of the team with the domain and the size of the project, there is a balance between launching straight into the code and taking some time to understand the overall view. For this purpose, storyboards, models, and the two in combination provide important tools in the developer's toolbox.



CQAA Events - Another Look

In April of this year, Cindy Glaser and Nancy Kastl of CQAA led a Lunch & Learn discussing important issues of working with Distributed Teams. Cindy Glaser has provided the article below reviewing some of the topics discussed.

Getting the Most Out of a Distributed Team

I hadn't had my coffee yet, so when I walked into my office one Monday morning to find a strange man sitting at my desk I assumed that it was I who was in the wrong place. The man seemed lost; I asked him what he needed and he said he wasn't sure. He explained that he was the new QA analyst from our office in India and was here to be trained. I wasn't aware we *had* an office in India, let alone one that was home to my newest co-worker, and especially that my team would be responsible for training the new team.

The next week was a scramble as we tried to find answers to myriad questions: What would this team be working on? Were we training these people to take our jobs or would they be supplementing our workload? Was development going to be located offshore as well? Who did this team report to? Could we give them assignments? Who would be responsible for training them, and in what? And how do we get our regular work done in the meantime?

Consequently, the first few weeks of that two-month trip were not very useful for our offshore counterparts. We were able to throw a training plan together based on a combination of assumptions and available documentation, but our colleagues were ill-prepared to function as an independent team when they returned to their home office, and it was at least a year before we could call them true contributors. That's not to say they were not skilled or capable, but the opportunity to bring them on board as seamlessly as possible was lost, as was a great deal of time and productivity, and we also lost some good people on that team to frustration.

We are all in situations where we are required to work with others who are not co-located, whether they be telecommuters, colleagues in a branch office, or located overseas. The reason might be cost-savings, creating a 24-hour workday, convenience to local resources or consolidation with other organizations. Regardless of the specific situation and the reason, the challenges of working with distributed teams are, at their core, quite similar. Techniques for managing these challenges, however, are as numerous as the organizations facing them and need to be carefully considered when determining the most effective way of working together.

Step 1: Plan Ahead

Too many organizations take a modular approach to incorporating an offsite team: plug and play. They don't make the effort to integrate the team into the existing environment. Offsite teams and individuals need to be managed, just as an onsite team needs to be managed. This begins by establishing a road map for bringing offsite employees up to speed (or getting more out of an established team). Start by asking yourself how your distributed teams play a part in your organization. Evaluate your needs and potential challenges with questions such as:



Another Look (cont.)

- What are the cultural facets of your organization and your team? What's the workload like, what's the pace like? Do you adhere to specific process guidelines, such as ISO or CMM? Are you using Agile? Are you on the bleeding edge of technology or are you still on OS/2?
- What are the logistical implications of the distribution? Is there a time change? What resources need to be available to both teams in terms of human resources, facilities, etc.? Which departments and disciplines are available to the remote team? How will documentation be shared? How will access to common applications and directories be shared?
- What tools are available to facilitate communication? Phone? Skype? WebEx? IM? How will these tools be used, and are they sufficient?
- Who does the remote team or individual report to? How do these individuals know what is expected of them, and how is that monitored and managed?
- If it is a new team, how will training take place?
- How do you define success? Cost containment? 24/7 development cycle? What is the expected result?
- What will the remote team or individual work on? Will they be working independently or on the same projects as the team at the home office, and how will time and geographic differences impact collaboration?
- What are the skills of the offsite workforce? Will there be language issues? What technical abilities are required for the job, and will people with those skills be available to you?

Your answers to these questions will help to establish some rules and routines (“the team leads will have a Skype meeting every Wednesday at 8pm CST”) and anticipate roadblocks (“we need time to get new hardware through customs”).

Consider training also. Your organization probably has a formal on-boarding process for new associates, yet it doesn't expect offsite folks to go through this process. Why? Also consider the method being utilized. Would documentation be more effective than a phone call? Would flow charts or videos be a better mode? Don't forget to train your onsite team as well; they are going to want to know what their roles are in helping their offsite colleagues ramp up, and how their work will be affected.

Step 2: Manage

This should go without saying, but you'd be surprised at how often this often doesn't happen (or maybe you know all too well). There are plenty of teams floating in oblivion, unsure what is expected of them and what role they play because they do not have a strong management structure (or the one they have is at odds with yours). You might have even heard hushed talk in the hallway about that one employee who works from home but no one knows what he does. Why would we hold offsite employees to a different standard than those who work in the home office?

Offsite employees need to be managed, just as your onsite teams need leadership and direction. If you have an impediment, you can walk over to your manager's office or report it at your daily standup. Do your remote employees have similar channels? If you have an issue with the performance of one of your direct reports, you can call them into a conference room and have a timely discussion. How can you have that difficult conversation with someone thousands of miles away, and achieve an effective result?



Another Look (cont.)

Make an effort, particularly at the outset, to make your expectations clear and follow through if they are not being met. Indoctrinate your associates into your culture, whatever that entails. Review performance in the same method you would use with your own team. This can be challenging, particularly for those who are not in management positions, or if your offsite colleagues report up through a different silo or are even part of a different organization. In this case, determine how the relationship between teams or individuals will be fostered and how performance is managed. Can the offsite management team conduct 360 reviews to get feedback from the people they are working with at the home office? Can the day to day relationship be managed or monitored by an on-site liaison? How will non-performers be handled? This is true in the case of telecommuters as well; they need to know how they will be evaluated and what your expectations are in terms of availability, meeting attendance, status reports, etc. Agile methodologies, SCRUM for instance, can be useful in this situation because they rely on daily communication of status and increased visibility, and there is a ready-made liaison in the Scrum Master.

Get involved in the interviewing and hiring process if possible, particularly if hiring is being done by offsite management or a consulting organization, or problems can start before the team is even in place. While managing a standalone QA team at a branch office in China, we experienced an incredibly high rate of turnover, which is common when working with some offshore workforces. Some of the new hires would leave after a few months or, worse yet, a few days. Management at the branch office insisted this was a product of the economy; there were plenty of competing offers available and new hires would leave if they could make an extra dollar an hour across the street. We weren't satisfied with this response. Surely workers in China are similar to us in that they are not motivated solely by money. Upon reading the job description that was being used to recruit new candidates we realized that it barely even referenced the fact that this was an IT position. Thus we were attracting candidates who had industry experience and aspirations but were limited in their IT skill and interest. Several days was all it took for a new candidate to figure out that this position was not a good fit. After rewriting the job description to entice the right people, our turnover problem was significantly reduced.

Step 3: (To paraphrase a real estate mantra) Communication, communication, communication

I had the privilege earlier this year to lead a workshop on the subject of distributed teams, and the vast majority of the pain points my colleagues brought up came down to communication issues. These issues broke down into the logistical (how do I reach this person?) cultural / language-based (is my message being received and understood correctly, and do I understand what is being communicated to me?) and interpersonal (how do you build relationships with people you don't see face to face?)

Approach these issues head-on; anticipating difficulties will mitigate their impact. Determine the accepted modes of communication between onsite and offsite employees and instruct your team to use these channels. Evaluate your methods regularly to see if they are effective. Engage your offsite employees in regularly scheduled conversations whenever possible. If you do regular one-on-ones with your staff at home, schedule them with your offsite folks as well. If this is not possible, at least engage in frequent discussions with a team lead or other representative at the remote location. This is especially true at the outset; once the team becomes more self-sufficient the need for these meetings is not as great (but make no mistake, it's still there).



Another Look (cont.)

A huge and often overlooked key to success is understanding and respect for other cultures. This doesn't necessarily mean US culture vs. Indian culture. It can also mean Midwest vs. East Coast, developer vs. QA analyst. Gaining an understanding of where your colleagues are coming from and establishing a baseline of trust goes a long way in building a working relationship, so find ways to do this with your remote teammates. You may not be able to travel to a remote location, but you can still initiate a casual conversation with an offsite colleague to determine what is important to him or her, and sharing what is important to you. (You are ostensibly already doing this with the people you work with every day.) Of course there are some limitations to this method because it's harder to pick up on nonverbal cues, but a little respectful whistling in the dark can help you establish successful avenues (and discover which ones to avoid). Sometimes something so small as sharing pictures or videos between teams helps to establish trust, it gives the teams an opportunity to put faces with names and get to know each other as human beings. Do some research on virtual teambuilding exercises that will allow each group to learn about the other.

I was involved with a group of American and Japanese analysts who were working together on a large testing project. The Japan team had clear expertise in the area being tested, but was completely off-base on the requirements. The project manager demanded daily status calls between the two teams to try to iron out the misunderstanding. Each call seemed to go well. The US team would communicate the requirements verbally to the Japan team, and then wait while the Japan team discussed the requirements among themselves in Japanese. After a few minutes, the Japanese test lead said "Ok." Assuming that they had successfully communicated the requirements, the US team thanked the Japan team and ended the call. The next day, it was clear that the Japanese analysts still did not understand the requirements. After going through this frustrating cycle a few times, it was clear the message wasn't getting through. When the requirements were laid out on the next call, the explanation was followed by asking "do you have any questions regarding what we just discussed?" This simple question prompted a more two-sided discussion regarding the requirements. The test lead followed-up the conversation with an e-mail confirming everyone's understanding, and the project was back on track.

Steps 4, 5, 6 and 7: Measure, Report, Correct, Repeat

This goes back to determining what is important to your organization, and what defines success to you and the people that you work for. Is cost the driving force? Keep track of how the distributed team is impacting cost, not only to the bottom line but to things like quality, customer service and retention. Is time a major factor? Measure throughput and adherence to schedules. Identify trends. It's expected that any culture change will result in some slowdown; the trick is to identify areas of inefficiency, correct them, and work to trend upward.

To that end, track performance using agreed-upon metrics. Again, these should be similar to the ones you use to track your onsite team's performance. Learn from mistakes. Improve processes, even if it requires trial and error. Establish baselines, track improvement or deterioration. Reward successes and hold the team responsible for failures. If all of this sounds familiar it's probably because your own work is being managed this way. Be careful, however, when comparing the performance of your onsite and offsite teams; take into consideration the relative experience of the team members and the impediments faced on both teams. Do not create a culture of competition (though it might be a useful incentive later on).



Another Look (cont.)

Now comes the hard part: communicating information back to your management team that may be negative. Your managers want this arrangement to succeed; their bonuses might depend on it, and they are going to want to see swift, positive results. News that “it's not working” isn't going to be well-received. Tracking metrics that support your assessment will help get your message across, as will a contingency plan that can reverse the trend of failure.

So what would this contingency plan be? Some strategies I've seen work are:

- Regular meetings with the team. If these don't seem effective, consider changing the venue. If phone calls lead to confusion, try IM or WebMeetings. Make sure the offsite people know that you care about their performance and consider them part of the greater organization.
- Site visits (if budget allows).
- Training in weak areas.
- Evaluation of individual performance. Don't generalize to the whole team the weaknesses of one or more members.
- Creating a career path and clear instructions on how to reach the next level. Your offsite employees want to climb the ladder too.
- Evaluation of hiring policies. Is salary competitive? How does the working environment compare to other organizations?
- Ensuring that your offsite folks have access to the same information and resources that your onsite employees do. Set up a knowledge base in a known centralized location, such as a Wiki, to store frequently asked questions and common terms that may be unfamiliar to the offsite group. Nothing is going to frustrate your onsite employees more than answering the same question for multiple people.
- Asking open-ended questions (instead of “will you be done with this tomorrow?” ask “when will this be complete?”); this avoids some cultural communication issues and might provide you with information you didn't even realize you were looking for.
- Arranging team building exercises that give your distributed workforce a common goal that they can work toward together.
- Identifying strong performers, both on- and off-site, and utilizing their expertise in leading the offsite team and training new employees.
- Identifying and resolving technical limitations.
- Changing the source. If your message doesn't seem to be getting through, try having someone with a higher rank in the organization deliver it.
- Evaluating task size. Creating and assigning smaller, more manageable tasks can help you identify potential problems early, and build a track record of short-term successes.
- Getting more involved in the interview process, asking specific questions relating to real issues and challenges that candidate will face, to vet out any misrepresentation or misunderstanding of skills.

Offsite teams and individuals are not wind-up toys; they require the same (and possibly more) hands-on care than your home-office team. Although these individuals are not physically present in your office, they have a significant impact on the success of those who are. Planning and setting goals at the front-end will ensure that you get the most out of your teams, regardless of their location.



Hot Topics

Software Quality and Testing Professionals face a number of issues as they attempt to provide value to their project teams and organizations. This section will present some issues shared with the CQAA Newsletter Editors. We have gathered a few questions/issues to start and will continue gathering ideas in the future. These topics will be addressed in future issues with a recap, as shown below, or perhaps a full article.

How do I combine Waterfall and Agile methodologies to effectively deliver software quality in my projects?

Should QA have the authority to stop a project from going live?

How do I know I am using the right methodology? Why does it matter?

Who decides “best practice?”

What is “just enough” testing?

Help us help each other. Have you solved one of the questions above or are you facing a similar issue? Please share what worked for you.

Send us topics you would like addressed in future issues or let us know if you wish to contribute to the article through an interview or by providing some content. Contact us at Editor@cqaa.org

CQAA LinkedIn Topic of the Month

A recent topic presented to the CQAA Linked-In community asked “**What technical skills (such as programming languages, etc.) do you require to be effective in QA?**” Listed below is a recap of the discussion.

- QA Analysts should have a good concept of programming languages and how they work.
- VB Script is a very popular language, however, other popular application technologies require more Java type programming skills.
- The most popular database languages are SQL and PLSQL.
- VBScript or Python can be used for backend system automation and also it can be used to create standalone scripts.
- Web Services (tools and skills) SoaTEST, SoapUI are also key knowledge for those application platforms.
- The bottom line to being an effective Quality and or Testing professional is a strong ability to produce effective test cases and test scripts.
- It's not the programming skills that discover anomalies in software, it's the individual's mindset and the desire to locate anomalies that's value adds to software quality.
- Any specific technical skills will likely be obsolete in a few years; a quick learner who is flexible and committed to using the right tool at the right time is going to be more successful in the long-term.



The Quality Journal

June 2011

CQAA Mission

The Chicago Quality Assurance Association, the Chicago Chapter of the QAI Global Institute is a nonprofit professional organization that was established in 1984 to promote quality principles and practices within Chicago-area companies. CQAA provides a forum for information professionals to present and discuss quality and process management within information systems, technology, and services.

Objectives

- Provide a variety of educational opportunities in the Chicago area for quality professionals and other advocates of quality.
- Facilitate networking and the exchange of ideas among quality, process management and information technology professionals.
- Sponsor presentations on quality and related topics by experts and by members.
- Foster professional certifications through access to examinations held in the Chicago area.
- Provide an opportunity to earn recertification credits towards professional certifications through membership and attendance at educational programs.
- Maintain lines of communication with other professional associations and foster cooperative activities of common interest.

Chapter Leadership

CQAA is served by a self-perpetuating Board of Directors that meets to plan, implement and review programs and functions. To ensure that the CQAA Board remains strong, the directors have adopted a succession plan. All board positions have detailed descriptions that identify the responsibilities of the Board Members. Board positions that become vacant will be posted on the CQAA website and applications will be accepted at that time.

Board Members

Cindy Glaser, Membership
Nancy Kastl, President
Mike Lawler, Secretary and Journal Editor
Kim MacDonald, Certifications
Fabrizio Stortoni, Programs
Sara Thomas, Treasurer

Upcoming CQAA Events

Date	Title
June 22	Exploratory Testing in the Enterprise
July 27	Mobile Technology Testing
August 9	Testing @ Microsoft
April 30- May 4, 2012	QUEST Conference