

---

<http://www.testingeducation.org/BBST/BBSTSpecificationTesting.html>

<http://www.testingeducation.org/k04/documents/specBased.pdf>

<http://www.testingeducation.org/k04/documents/specBased.ppt>

---

**Vijay:** Who are the participants of a specs review process?

**Dr. Cem:** The participants of a spec review process include all of the reviewers and no one else.

**Vijay:** Suppose I am working for a product based company that is launching a product which is a new one.. The developers, testers, moderator, reviewer (part of the review process) do not have a previous idea of the same. How will they review the specs?

There are no previous products of that sort in the market.

**Dr. Cem:** The videos attempt to answer exactly this question. I am not sure what you do not understand

---

**Ashish:** My understanding about specification document is that it is a live document. Is this correct. Can there be cases where spec documents get freeze and no changes are allowed in that

**Dr. Cem:** we can freeze any document any time

**Ashish:** So will it be wrong to call Specification document a live document

**Dr. Cem:** It is not right or wrong to call a specification document a live document. A document that is open to revision and is in fact being revised is a live document. This includes some specifications. A document that is no longer being revised is not. For example, when someone commits commercial fraud and ends up in court, their product specifications might be evidence in court. The specifications we take in evidence are not live documents

**Indra:** Once we Freeze, but over a period of time if we need to append new requirements/enhancements then do we need to write a new spec or we can use the old one?

**Dr. Cem:** Several of you are asking questions about the process associated with writing and revising the specification. That is context-dependent. Each development group or company does what it does. Every variation you can imagine has been tried. People can freeze and unfreeze. People can write competing specifications, etc.

**Indra:** It is impossible to fully specify anything that is complex. So you might be wishing for something that is impossible@Cem: How could it be minimized....?

**Dr. Cem:** are you writing to me as a tester or as a project manager? The roles are very different?

**Indra:** As a Tester.

---

**Dr. Cem:** my experience as a tester and as a test manager is that project managers do what they do. I am not their manager and I do not control their processes. On different projects, detailed specifications are more valuable or less valuable. As a project manager on some of my most successful projects, I drafted many of the specifications in face-to-face meetings with my lead programmer and we scribbled them on napkins (at a bar). It worked for us. It didn't work so well for the testers, but it worked for the project...

**Dr. Cem:** - as a tester, you are one of many project stakeholders. What is convenient for you is not necessarily what is best for the project. It is possible to over specify a product. In fact, in my experience, this is common. Imagine you are a designer (or a project manager who owns the design of a small product) and you do not fully understand what your stakeholders want. You might choose to write down many details, but most of these details are probably wrong. If you treat this as a draft, you might get corrections in discussion with your stakeholders and those corrections might come in the form of modifications to mocked-up screens or in the form of changes to code as you program the UI together or in the form of meeting notes. You might or might not translate those back into a written specification. You might keep working with the stakeholder on successive drafts of the prototypes or program instead, or you might keep working with a stream of emails. The tester might not see these interactions and be confused, and that means she has to work harder, and develop her own expertise on what a product of this kind should do or could do, but the project might be progressing very nicely and the quality (in the eyes of the stakeholders who will use and pay for the product) might be very high

**Indra:** thanks for the clarification.

**Maik:** @Cem: regarding your answer to Indra... for me it makes sense to involve the testers in early stage of specification creation.

Maybe not the tester, but the test team leader or manager

**Javed:** @Maik: I agree testers must be involved during specification process because many bugs, improvement suggestion and understanding of project comes through this process

Is it advisable to have one \*master\* specification, which is frozen and handle all following changes as future change requests? Or is it better to have a live document (with all time/resource headache)?

**Dr. Cem:** as a TESTER, I have never controlled the format or the source control of the documents of the development group that writes the specifications. As a project manager and development director (and as a lawyer), I have kept specs in different ways for different purposes, but as a tester, I get what I get. On projects that have multiple versions over year, my experience is that there is no master specification. I discuss this on slide 11

**Dr. Cem:** @Maik you wrote "for me it makes sense to involve the testers in early stage of specification creation" -- I agree that it sometimes makes sense to involve testers in the early stage of specification creation. However, if I am developing a product and the test group is 2000 miles away and will join the project next month, it is not possible. And if the tester who would join the project is going to talk about

process models and demand more detail than my client and I want to create, I will keep the tester out of the room. The tester is a service provider to the project. The project manager may or may not know how to use this service effectively -- (a) some people who own cars have knowledgeable and effective relationships with their mechanics, (b) some people drive their cars into the ground instead of getting preventative service, and (c) some people are always cheated by their mechanics -- the project manager may or may not know how to work effectively with the testing service provider, but ultimately, the management of the project is determined by the manager of the project. If you can genuinely help the project manager control the project -- and when I say this, I include in my meaning that the project manager has to agree that she thinks you are helping and she values your help -- then you will be included in many aspects of the product's development that other testers might not be included in.

**Maik:** @Cem: Thx for the answer. I agree (especially since c) is me with cars^^) in most parts. I would still try to convince the project/product manager to involve testers earlier. Maybe after client/project meeting, but before dev/testing starts... to review and improve specs. If project manager is "experienced", he would involve testers himself. If not, it should also be part of our professional testing role to try to help him. How to do it diplomatically is another topic...

**Dr.Cem:** @Maik you wrote, "I would still try to convince the project/product manager to involve testers earlier" -- in my experience, testers build trust and respect with project managers over time, and their role expands as trust and respect build. We earn our privileges and our scope. A challenge for some people in the testing community is that they work in independent test labs and they therefore work with different project managers in different companies every time. This is extremely difficult on all sides -- even when all of the work is local -- I used to outsource some testing from San Rafael (a little city north of San Francisco) to test labs in Sunnyvale and Palo Alto (little cities just a few miles south of San Francisco, an easy drive from San Rafael). All of the problems that people talk about when they complain about outsourcing testing from North American or European companies to India showed up when we outsourced only 30 miles away. Building trust, respect, and appropriate domain expertise across a corporate barrier is very difficult and too complex a discussion for today, I think. But when some people talk about the need for detailed specifications, those people are often remembering the pain of outsourcing.

---

**Jassi:** so does this mean that our entire testing would be focused only as per the specs given?????

**Dr.Cem:** I have never seen a complete specification, and I believe it is theoretically impossible to fully specify a complex product (just as it is theoretically impossible to fully test a complex product and impossible to fully specify any natural language). Therefore, it would often be unwise to focus the entire testing of a product against the specification.

---

**Javed:** @Cem : currently I face in my project in Specification client never describe UI feature but when deliver the product he/she complain lots of UI bugs so how should overcome such kind of situation

**Dr. Cem:** -- it is common in development for specifications to be incomplete. It is impossible to fully specify anything that is complex. So you might be wishing for something that is impossible

**Javed:** Why? There should be some standard for UI.

**Dr. Cem:** whether there is a specification or not, the user interface must evolve as the client understands what is needed. Therefore, I try to engage my customer base as I develop the product, building prototypes and so on.

**Javed:** so I just want to know how should I follow to test UI so that minimum bugs found by client side

**Dr. Cem:** -- standards for UI -- there are standards for the UI, but they cover the superficial aspects of the UI. Are your clients complaints the kind that would be covered by a standard, or are they more application-specific?

You always approach every task from yourself. However, your opinion is only as good as your knowledge and care...

**Javed:** He/She complains both kinds of bugs but maximum belongs to application-specific

I have tested many applications but I always face such kind of situation.

Another problem is I always create test cases on the basis of specification to minimize high risk in application but in spite of that some major bugs identified that are beyond the specification...so what your experience says about this

**Jawad Zafar:** I don't think that each and everything could be documented. We have to look a little beyond that is written.

**Dr. Cem:** -- @Javed, @Indra, First, no one can ever assure a complete, bug free product. Even if the product works perfectly according to your intention, quality is in the mind of the stakeholder and stakeholders differ. What is bug-free to one is not-so-bug-free to another.

Second, a weak specification is only a failure of the development process if the development process needs a stronger specification.

**Dr. Cem:** @Javed -- specification-based testing is only one approach to testing a product. It is useful to test a product against every claim that is made about the product (this is specification-based testing). However, there are many other techniques beyond the specification-focused techniques. There is no reason to follow only one approach in testing

**Javed:** @Cem actually my intention is to find those hidden bug that are causes of bigger loss in business so how we can uncover those bugs

**Dr. Cem:** @Javed, I am not sure what situation you want me to overcome. The fact that there are bugs that are not anticipated in the specification?

**Dr. Cem:** @Javed, you wrote: "actually my intention is to find those hidden bug that are causes of bigger loss in business so how we can uncover those bugs" -- let me suggest that many of the most serious bugs in the product are in areas that were not well designed or carefully considered by the project team. Therefore, testing against the specification is the best way not to find these.

**Indra:** @Cem: Can you plz let me know about this...

**Dr. Cem:** testing against the specification is the best way not to find these.

@Cem: Does it not affect the Product/Project? As a Tester we must cover the as much as we can before the end user experiences the weird behavior in an application? Is it NOT?

**Dr. Cem:** @Indra: you wrote, "Does it not affect the Product/Project? As a Tester we must cover the as much as we can before the end user experiences the weird behavior in an application? Is it NOT?" I do not understand how your question relates to my assertion that specification-based testing is a poor way to uncover the kinds of problems that were not anticipated by the programmers. I am not suggesting that testers should not test for serious problems. I am suggesting that specification-based testing is effective against some risks and ineffective against others. This is true for every family of test techniques. This is why I prefer not to restrict testing, on any project, to only one family of test techniques. Does this address your question?

---

**Catalin Anastasoie:** @Cem: You've said that your most successful projects had their specifications created in 1 on 1 meetings or in talks with the development team - using this as a starting point are you saying that we as testers should try to be more communicative with the developers rather than following processes?

A relation similar to the ones that you had with the lead developers from your top successful projects?

**Javed:** @Anastasoie as per my view communication with developer is important more important for testers to communicate with the client create mapping between client argument and developers as well as understanding of domain of the project

**Dr. Cem:** You asked "we as testers should try to be more communicative with the developers rather than following processes?" When it is possible for the testers to interact more with the other developers, this is desirable. Formal processes are valuable when all other methods of communication fail. They are also valuable when there is a risk of litigation (if your product can kill someone, it is good to have formal records...) But highly detailed, formal processes are a very poor substitute for more interactive communication when it is possible

**Catalin Anastasoie:** @Cem: Do you have or are there some heuristics that can be used as an example?

**Dr. Cem:** @Catalin -- you wrote, "@Cem: Do you have or are there some heuristics that can be used as an example?" In this multithreaded discussion, I have lost context. I apologize. As an example of what?

**Catalin Anastasoie:** @Cem: My last question is related to what testing technique is better for a given project?

**Dr. Cem:** @Catalin: You wrote, "Cem: My last question is related to what testing technique is better for a given project?" I apologize, but this is too far out of scope for this session. I am happy to have this discussion, but it is a many hour discussion, not a three-best-practices-and-you're-done. (If you want the three-best-heuristics, ask istqb.)

---

**Uday Vussainsagar:** @Cem: Does domain knowledge plays a key role in testing spec-based?

What if a tester needs to review a spec which he is completely new to domain? How can one actually find the defects?

**Dr. Cem:** @Uday you wrote, "What if a tester needs to review a spec which he is completely new to domain?" This is a common situation. Much of this video is an effort to address this--you are trying to figure out what the specification covers, and then figure out what it means (all of the material on active reading is about figuring out what the specification includes and means). From there, you can assess the specification for clarity and logical consistency--but you also have to go study up on the domain to assess the specification for suitability. You have to try the product out, or try out prototypes or create your own prototypes and your own models based on the specification to see whether they yield something that is useful. You might have to ask other people a lot of questions...

**Javed:** @Cem for example y2k plm faced in 2000 this is a big example but same in banking domain ,healthcare domain, chemical domain during creation of chemical raw material

**Dr. Cem:** @Uday -- domain knowledge can play a key role in all types of testing. Note that every product involves many domains. For example an ecommerce application (imagine testing amazon.com) involves understanding user interfaces and user interaction on websites, credit transaction processing and related laws, shipping, inventory control, output devices (e.g. printing)--detailed understanding of any of these will influence your testing

**Uday Vussainsagar:** @Javed: What is the use of testers if the tester does not have domain knowledge?

**Javed:** @Uday then you have to acquire domain knowledge if you are not getting then you have to take initiative and communicate with your developer and client

**Uday Vussainsagar:** @Javed, Cem So the bottomline is that domain knowledge is very important to have a say at spec meeting

**Javed:** @Uday in Specification meeting if you are participating then customer expect some inputs and suggestion from your side as per my view its needed

**Javed:** @Uday this is the only way to get domain knowledge as customer point of view and test the product as per his expectation

**Dr. Cem:** @Javed -- you said, "this is the only way to get domain knowledge as customer point of view and test the product as per his expectation" -- Much of the most important domain knowledge is not from the customer's point of view. For example, my assessment of the performance characteristics of a product, of the product's compatibility with third party devices, of its compatibility with third party protocols (think of ecommerce systems sending messages to MasterCard and Visa, asking for credit authorizations--the credit card companies created the communication protocols, you just use them), my assessment of the effectiveness of the product's defenses against third-party efforts to spy on the data stream or modify the data stream -- all of these involve knowledge of technical domains that the customer probably knows nothing about.

**Javed:** @Cem right so we shouldn't always depend on customer

**Indra:** @Javed, Cem So the bottomline is that domain knowledge is very important to have a say at spec meeting@ Uday: In Real time, it's not POSSIBLE to have domain knowledge at each level like dev/tester etc....but I don't know how to take steps to improve this.

**Dr. Cem:** @Indra -- You mentioned, " In Real time, it's not POSSIBLE to have domain knowledge at each level like dev/tester etc" -- I agree. When I build test groups (as a test manager or later when I consulted to test managers and executives) I prefer to hire a very diverse staff. For example, when I consulted to ShareData (a company whose software managed employee compensation programs involving stock options), we build a test team that included (a) a test manager who had a thorough knowledge of general business processes (this was jack falk), (b) a programmer who had a thorough knowledge of implementation, and implementation risks of tax-related software (the most complex issues in stock-option-related systems are tax issues), (c) a lawyer who specialized in tax law, (d) a general-purpose tester (a fast learner who liked to break things). We also interacted closely with the technical support group (many of them were product experts) and with system designers who were Certified Equity Professionals--we reviewed each other's work and cross-trained each other. Different people were better at assessing different risks

**Maik:** @ "Domain knowledge"... could it also not lead to the common blindness of being to \*close\* to an area? and that way causing to skip or otherwise influence testing? In opposition to a tester who has no domain knowledge (or lesser) and he will test in a different way?

**Javed:** @Maik AS per my experience I am testing chemical product application and I have very little knowledge it doesn't mean if you don't have domain knowledge then u lead to wrong way it your perception or view of application how you have developed the plan to test this product at that time you have to make requirement traceability matrix to make product reliable and slowly you will also get the idea or way to test this product

**Uday Vussainsagar:** @Everybody...For example, To test any product related to drug submission to FDA, a tester mandatorily should have knowledge on the drug submission process...I guess before a tester goes for a spec meeting he/she do a proactive study on the domain

**Dr. Cem:** @Uday -- you wrote, "To test any product related to drug submission to FDA, a tester mandatorily should have knowledge on the drug submission process." Let me suggest instead that to test any product related to drug submission to FDA, SOME TESTER ON THE TEAM SHOULD mandatorily should have knowledge on the drug submission process.

**Dr. Cem:** @Indra, Javed -- no one is responsible for my ignorance of quantitative trading systems but me. In the United States, over half of the stock trades are now driven by automated trading systems rather than by humans. If I want to understand how the market works, I have to understand the trading system. If I want to test models of the market, I have to understand the range of trading systems. If I want to test the models underlying a specific trading system (one of the things that distinguishes a high quality trading system from a low quality one is whether it makes money for the people who use it, yes? So assessing the quality of the system requires assessment of its probable value to users, which is often done in highly-detailed simulations)--If I want to understand these, I have to educate myself. If I am unwilling or unable to educate myself about these, I should test something else. If I am more generally unwilling or unable to educate myself about a product domain, its risks, and how to assess a product in that domain, I should give up on testing and do something easier that requires less learning, like programming.

**Uday Vussainsagar:** @Cem...I have another question? You are talking about what is the value of writing a test plan. So, if we can understand clearly what is written in the specs...Can we get away with test plan? Can you give instances as when to write a test plan and when not to write?

**Dr. Cem:** @Uday -- You write, "I have another question? You are talking about what is the value of writing a test plan. So, if we can understand clearly what is written in the specs...Can we get away with test plan? Can you give instances as when to write a test plan and when not to write?" -- can you remind me where in the discussion of spec-based testing (the video or slides); I say anything about the value of writing a test plan?

**Jawad Zafar:** @uday: Test plan would contain each and everything as per the Testing, while specification based testing is just a technique

**Maik:** @ Test plan... it is also a living document... so if new information comes, it should be adjusted

**Jawad Zafar:** @uday: Test plan would contain your objective, scope, test strategy, test environment details, assumptions, exclusions, bug mechanism, acceptance criteria, shipment assurance criteria, regression, total cycles etc.....

**Khair.un.Nisa:** @Jawad: it works well if things are explicit and still we find it confusing somehow. If things are implied, there is a high probability of skipping things....

**Jawad Zafar:** @Maik: yes, you should convince your management with logical reasoning to support your view

**meeta prakash:** @khair.un.nisa - how do u know that what is implicit and what is explicit in completeness?

**Jawad Zafar:** @javed: one should always present customer with the options that he could choose from. This would facilitate him

**Khair.un.Nisa:** @Meeta: When it goes in production we only then come to know that some information was implied...

**Jawad Zafar:** @ Nisa: a tester should always behave as a client. He should look into the product with the eyes of the client and the target audience, for which it is being made

**Dr. Cem:** @JZafar: You write, "A tester should always behave as a client" Always? What about the tester who pairs with the programmer to develop good unity tests? Or the tester who imagines how to operate as a thief in order to circumvent a system's security? A tester should adopt whatever role or mindset will help him explore the issues that the stakeholders need explored.

**meeta prakash:** @khair.un.nisa - but we are testers ....how much was "some"?

**javed:** @Jawad yes that's why I have said during specification testers must be part of that meeting or after immediate preparation of specification should be go through the testers so we can find bugs and suggestion earlier stage

**Uday Vussainsagar:** @Jawad so the test plan is always different from the spec based testing

**Jawad Zafar:** @ uday: in this, the strategy part would be related to the specification based test strategy that we are going to imply

**Khair.un.Nisa:** @Jawad: yes sure provided we act as analysts too..... what we get are concrete requirements. If not, we highlight them and get answers for it. If a tester has real good knowledge and experience of the domain, only then the tester can highlight such issues. But think.....we always are not well rehearsed with the environment in which the product will work. We usually have high level information only

**Jawad Zafar:** @ uday: in our company, our test manager forces upon the Pm that all the emails should be cc to the test team, whatever they may be. Moreover, the FS/SRS is prepared by the dev, but it is reviewed by the test team

**Dr. Cem:** @Maik, @ Khair -- you are assuming that the test plan has value, that a test plan has more value if it is written in detail, and that there is value in keeping the test plan updated. You realize that these are all assumptions, which are probably correct in some contexts but which are incorrect in others, yes?

**Maik:** @ Test plan. Yes, all testing plan is "planning" and therefore a more or less sophisticated guesswork

**Dr. Cem:** @JZafar: You write, "A tester should always behave as a client" Always? What about the tester who pairs with the programmer to develop good unity tests? Or the tester who imagines how to operate

as a thief in order to circumvent a system's security? A tester should adopt whatever role or mindset will help him explore the issues that the stakeholders need explored.

---

**Parimala Shankaraiah:** @Cem: How to get close to the stakeholders' expectations given that testers are kept in the dark in most interactions about the product (I understand this is a very broad question)

**Dr. Cem:** @Parimala: You wrote, @Cem: How to get close to the stakeholders' expectations given that testers are kept in the dark -- Let me address this with an example. I am currently learning how to test financial models -- the kinds of models that professional stock traders use -- I have been working on this for the past year. It is interesting to me that the kind of testing involved is an excellent illustration of intensely automated exploratory testing. On my desk, books that I looked at yesterday, are Bernstein's "The Ultimate Day Trader", Narang's "Inside the Black Box: The Simple Truth about Quantitative Trading", Gunn's "Trading Regime Analysis", Katsanos' "Intermarket Trading Strategies" and Tortoriello's "Quantitative Strategies for Achieving Alpha". I am finally getting sophisticated enough that experts in the field are willing to talk with me. In addition, yesterday, I worked on investopedia.com, scottrade.com, interactivebrokers.com, morningstar.com, marketgrader.com, and several other sites. No one keeps testers in the dark. Testers keep themselves in the dark.

**Javed:** @Cem but I as per my view rather than keep themselves in dark, developers are more responsible for that...What do you think?

Because developer always passes incomplete information to testers ... :)

**Dr. Cem:** @Javed -- you wrote, "As per my view rather than keep themselves in dark, developers are more responsible for that ...What you think?" I think people often make excuses for their own ignorance and blame others. No one will ever give you complete information about anything. It is up to you to learn what you need. In my experience, I had to do a lot of that learning on my own time, at my own expense.

**Indra:** In my experience, I had to do a lot of that learning on my own time, at my own expense.@Cem: Sounds promising and enlightening. Thanks.

---

**meeta prakash:** @Cem ...one typical issue we face is that the testers start running towards the result (Pass/Fail) rather than pursuing and questing the work given out as testing bit. How can we change the mindsets ...Any suggestions? This also leads to darkness as there is hardly any urge to learn the domain details

**Dr. Cem:** @Meeta: Hi, Meeta, it is good to see you again. You asked, "One typical issue we face is that the testers start running towards the result (Pass/Fail) rather than pursuing and questioning the work given out as testing bit. How can we change the mindsets ...Any suggestions?" -- I am trying to understand how to tie this question to the specification-based testing material. It would not be helpful for me to write a general rant here. Can you help me by focusing this question a little more tightly?

---

**Catalin Anastasoie:** @Cem: As you've said Specification Base Testing is a testing technique, one from a large number of testing techniques. How can I assess which is the best testing technique to be used for a given project?

**Dr. Cem:** @Catalin: You asked "How can I assess which is the best testing technique to be used for a given project?" That is one of the defining questions of context-driven testing. There is no simple answer.

---

**Ashish:** @ Cem Kaner: For agile processes do we follow same rules for specification based testing

**Dr. Cem:** @Ashish, you asked "For agile processes do we follow same rules for specification based testing" -- I no longer know what anyone means when they talk about "agile processes"

---

**Khair.un.Nisa:** @Cem: Every specification needs to be documented explicitly and not implied in specification based testing? If no, we might skip some important scenarios that are ambiguous. How to catch them? (E.g. discussions only? asking directly from client)

If yes, who will take responsibility of it? Some clients are not technically strong enough to explain whole scenarios unless we question them.....and how important is it to highlight this risk?

**Dr. Cem:** @Khair, you wrote, "Every specification needs to be documented explicitly and not implied in specification based testing?" This is like saying that every day needs to be sunny and warm. It is not going to happen, and even if it did, your crops would fail for lack of rain. (In product terms, your product would wither for lack of creativity.) You cannot rely on specifications for all information; you cannot rely on specification-based testing for mitigating all risks. As to gaining information from clients, sometimes testers are allowed to work with clients. When I was a "senior software engineering / human factors analyst" at Telenova (a telephone company years ago), most of the programmers and most of the testers went out into the field for at least 1 week per year and several went into the field for up to 4 weeks per year (a field visit might be a week at a beta test site, watching customers working with, and cursing, our system). This was a very enlightened style of engineering. When I was software development manager at Power Up Software, we did nothing like this. To come to understand my market, I had to take a job on Saturdays at a store called Egghead Software (the Egghead chain was the biggest reseller of our products). I sold our competitors' products to people in our market, at the store,

to understand what made these products so special. I did this for almost a year. I also hired some Egghead managers to become testers :)

---

**Allmas Mullah:** Cem, it is common to see different people interpreting the same thing in different ways.

Especially while interpreting specs. What contingency plan would you suggest?

Would any of the following options be appropriate, if yes, which is most suitable; Peer review, business analyst review, client review.

**Dr. Cem:** @Allmas, you wrote, "it is common to see different people interpreting the same thing in different ways." Any natural language specification is ambiguous. Some languages are more ambiguous than others. English is a stunningly ambiguous language (a wonderful book on this, and on training people to recognize and deal with ambiguity in English is "Saying one thing, meaning another", <http://www.amazon.com/Saying-One-Thing-Meaning-Another/dp/1888222107> . You point out that there are different types of review processes for exposing ambiguities in specifications. You are correct that having different people in different roles can help. Also, having someone who is skilled at exposing ambiguity can help -- this might be more effective in practice, sometimes, than multi-person reviews. Another source of different-people-interpretations is that different stakeholders want different things from the product and often these preferences are incompatible. People will often read the same description differently, in a way that interprets the description in accordance with their preferences. Active reading processes can help people build scenarios -- examples -- that illustrate their interpretation of the meaning of the language. The tester can create scenarios and show them to different stakeholders who have incompatible views. This can expose ambiguity in a way that is compelling, and that would not be recognized by an individual stakeholder reviewing the specification (because she would interpret it as compatible with her needs)

---

**Javed:** @Cem in future I am going to test CRM,SCM product .they are vast project what would you suggest me to test such kind of product and I am also want to automate some part. These are very new to me

**Dr. Cem:** @javed: You wrote, "I am going to test CRM, SCM product .they are vast project what would you suggest me to test such kind of product" -- I apologize but you are asking too general a question for this session. This session and the next one or two are about helping you build a few skills in one family of test techniques. These would probably be one class of tool that you would use for such testing. The broader questions, what other techniques could you use, how you could manage them, what kinds of paperwork would you find useful, those are all many hour or many month issues in their own right.

---

**Shaham Yusuf:** @Cem -- My project has about 1200+ pages of Specs. It's a negotiated contract. Earlier we used to have 2200+ test cases. But now we have made all the predefined test cases obsolete and implemented exploratory testing in the project, crossing lots of challenges. We are using high level scenarios instead of test cases. These scenarios are open ended and do not have explicit steps or an explicit expected results. We use several oracles in our testing, one of which is the Specs. We also use check list for certain items of the Spec which are explicitly and clearly specified, e.g., date format, color etc. We record the execution through 'session tester' tool. We write the test designed and executed during execution, test data etc. This execution log along with the log of the executed checklist becomes a supporting document for any future or client reference.

Though we do all these things, it is practically not possible to capture all the tests designed and executed during testing. What I mean is due to time constraint it is not possible to record small-small things. We record what we feel is important to be narrated in the context of the release/iteration.

So question is, how do we make sure that we tested all the items of this 'Mega Specs' in each iteration?

**Dr. Cem:** @Shaham -- 1200 pages is not such a long specification. Now, you are saying that this is a negotiated contract. If that is true, then I assume that the funding for testing is also part of the contract and that the acceptance test process is part of the contract. The person who is paying for this product, the person who has signed the contract, probably has an opinion about what evidence is acceptable to them of coverage of the specification. That is the evidence that is often the most important to generate in this kind of situation. The other coverage question (risk coverage, rather than specification coverage) is a separate one. With the person who contracted for your services pay for that AS WELL AS for contract coverage?

**Shaham Yusuf:** @ Cem- The test approach (scripted or exploratory) was not mentioned in the contract. My question was more towards, specification coverage. If I put my question in other words; "How we can do exploratory testing, while maintaining the specification coverage"

**Dr. Cem:** @Shaham -- exploratory specification-based coverage. This is certainly achievable in non-contract testing. But you are trying to document coverage for an external person who is paying you to meet a contract. Somehow, you have to document your coverage. And you have to document it in a way that he will accept. This is a negotiation problem at least as much as a technical problem

---

**Maik:** @Cem: Finally finished the videos. By the first impression "active reading" plays a vital role for spec based testing (and other areas as well). My question: Besides the books and self practice... do you have any pointer, if and how such reading skill can be incorporated in daily work? Is it possible to find an approach, where the employees can be trained/educated...say 30 minutes a day?

**Dr. Cem:** @Maik -- you asked, "Do you have any pointer, if and how such reading skill can be incorporated in daily work?" -- A funny thing you should mention that. The proposal I have suggested for Weekend Testing is a three-part process. Day 1 is this video review. Day 2 involves getting some concept

mapping software, trying different products out and comparing them. Day 3 involves working with a specification (someone, not me, needs to supply a specification), using a concept map based on the Heuristic Test Strategy Model, to analyze the specification. James Bach and I used to find that people could do this kind of analysis well for maybe one or two hours a day (the hour sessions applying HSTM were part of the precursors of session based test management as a strategy). So, my answer to your question is that if you develop skill with this kind of tool, and with the use of it for structured analysis (such as, applying HSTM), then you can train staff to use it. The more they do it, and the more effective and encouraging your feedback is, the more they will use it in the future.

**Maik:** @Cem. Thx for that... so now u got me totally hooked to next sessions. ;) looking forward

---

**Dhara:** Dr.Cem can you recommend some of good sites that I can learn about banking/financial applications

**Dr. Cem:** @Dhara: "good sites that I can learn about banking/financial applications" I don't know much about banking. I know a bunch about investment modeling, but not banking. If you are wondering about investment software, the books and sites that I mentioned above have been useful to me.

**Dhara:** thank you Cem, I had this question whenever I came across new testing roles that emerge in banking sector

**Meeta prakash:** @dhara - do a google on various banking modules and u'll find very good ones available

**Dhara:** especially when they say about specific knowledge for banking software/tools, I really need to acquire that knowledge

---

**Vijay kalkundri:** @Cem: How is this topic relevant in my context? We are release testing team and we are not shared any of the test specifications? The system testing is done at other location... What should be our approach? We just go by our personal experience?

**Dr. Cem:** @Vijay -- if you have no specifications, you do no specification-based testing. Actually that is not true -- your product probably is subject to a wide range of implicit specifications and you might find it useful to test against them. Beyond that, when you ask "We just go by our personal experience?" -- different people mean VERY DIFFERENT things by that. How much personal experience do you have and how did you get it? You might need to do some research (reading, talking with other people, working with competitive products or products that will interact with your class of product) to gain the experience and knowledge necessary to competently test this product

**Dr. Cem:** @Vijay -- part of the answer to this is to ask what kinds of bugs you are missing or what kinds of dissatisfaction there has been in the field with the products you release. Imagine that every bug you

do not discover reflects a type of ignorance in your testing. If your experience does not stretch to make it easy to discover those bugs, you need to learn new things. If your experience does stretch, why are you missing them? What is the other constraint on your testing?