In the 60s and 70s Sweden's best ice hockey goalkeeper was Leif "Honken" Holmqvist.
He played when Soviet Union dominated, so many pucks went into goal, but he also did a lot of saves and was extremely popular.
He used to say "the posts are my best friends", which I have carried with me all my life.
The saying indicates it's about luck, but there's much more than good fortune to it.

It's the same with testers; you and your colleagues stumble on important things, and one might say that good testers often are lucky.
But it's not only luck, and that's what we will investigate in this presentation about serendipity.

Serendipitous Life

- Childhood
- Friends
- Education
- Job
- Family

- *It's about creating opportunities for good luck*

LearningWell

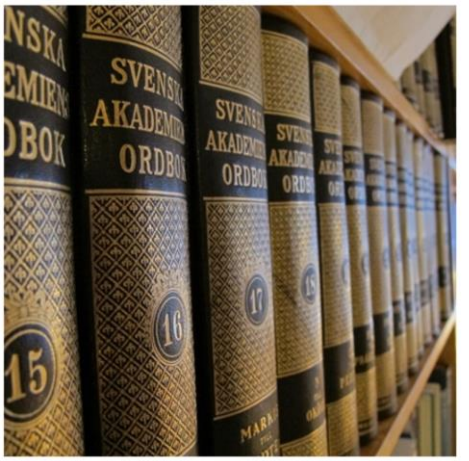
I had a happy childhood, pure luck!
I had great friends, where I still hang around with several of them.
I studied Philosophy at the University, without knowing the enormous benefit of the critical thinking skills it gave.
I wanted to be a software developer, but grabbed an opportunity to start as a tester, and never looked back.
I met my partner when I stopped looking for one, and we have three wonderful children, I have no clue how they became that…

A lot of things in life happens by chance, and my best trick is to create a lot of opportunities to get lucky.

Serendipity Definition

- "finding something valuable when looking for something else, thanks to an observant mind

- Serendipitet
- Serendipität
- Serendipia
- Sérendipité
- Serendipo
- Serendipisyys
- セレンディピティ
- سرندیپیتی
- Juhuslik avastus



LearningWell

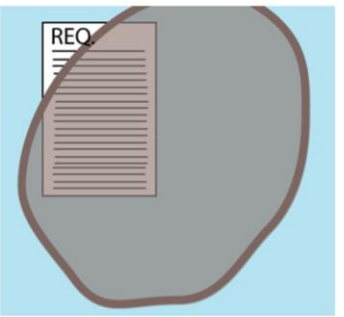
Serendipity happens when you look for something, but find something else, that is valuable.
It has been voted as one of the most difficult words to translate, which can be seen on this list.
Estonian is last on the list and make a good effort, but they haven't reached the top in ice hockey either.

The most famous serendipity example is when Fleming's experiment was a disaster, but when he looked carefully at the distorted results, he discovered penicillin.
He had an observant mind, so he could notice that the bad experiment had results worth looking deep into, and he knew enough about the subject, so he could understand that the results were important.

Sampling & Serendipity

- We can't test everything, we sample.
- We can observe carefully.
- We want to make many, rich tests.

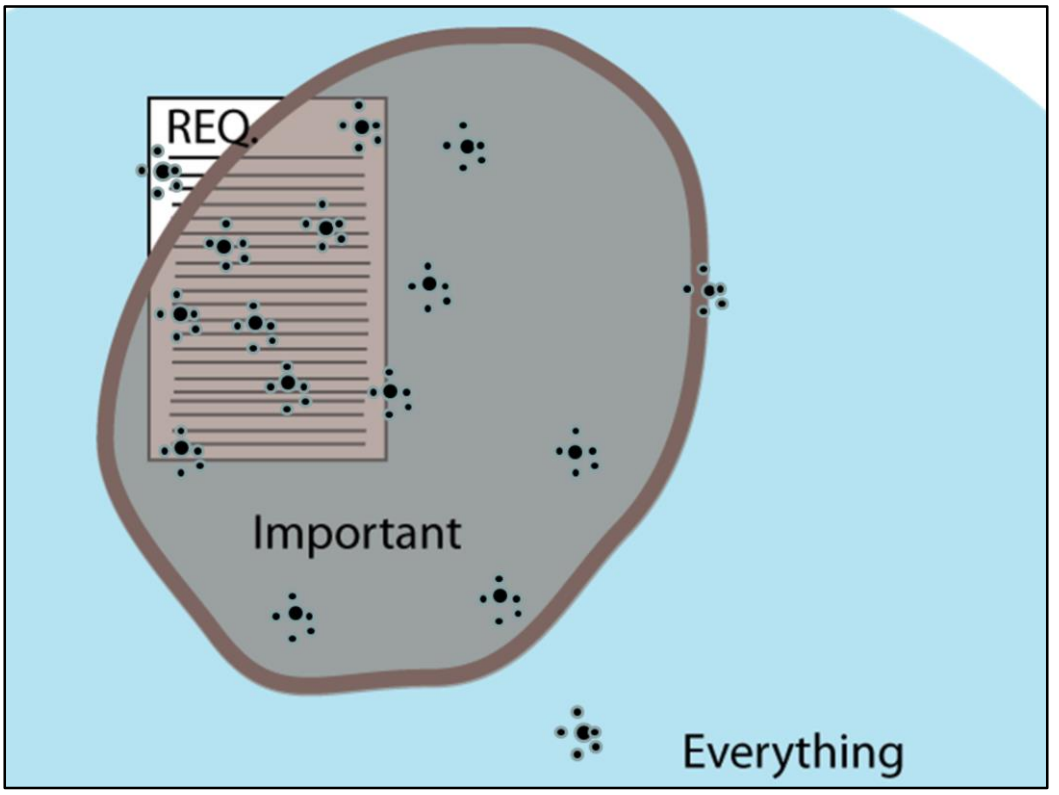
- We change sampling strategy as we learn more.



LearningWell

We all know we can't test everything. Testing is in the sampling business.
But as we test, we can be observant and notice more than we are looking for.
We can perform many, many rich tests in order to find more valuable information about the software.

We also want to change sampling strategy as we learn more, in order to learn even more.
Let's enlarge this image so I can explain more.



This is a brown potato. It is my simplified view on what software is for an investigating tester.

The square symbolizes the features and bugs you will find with test cases stemming from requirements/user stories etc. (that can't and shouldn't be complete)

The blue area is every possible usage, including things that maybe even no customers would consider a problem. This area is probably infinite, if we include the users' data, environment, their needs, different kind of sequences etc.

But we are lucky, not everything is important. So the brown area, the potato, is what is important, there lies those problems you'd want to find and fix.

There's where we want to do most of our tests.

Now if we do very exact testing, hitting one pixel at the time, we will find some stuff, surely.

But if we add serendipity to the mix, my theory is that we will see more than one pixel at the time.

We will be lucky, more often.

But this is not easy, thank God!

Testers need to learn a lot from many different sources, combine things, look at many places, think critically and design tests (in advance or on-the-fly) that will cover the important areas.

Yes, some part is luck, but there is a large portion of hard work, and a lot of testing-wiseness as well.

And my main point in this presentation is that serendipity is working to our advantage, we need to use it.

Prepare for Serendipity

- Error-Prone Machine
- Background Complexity Heuristic

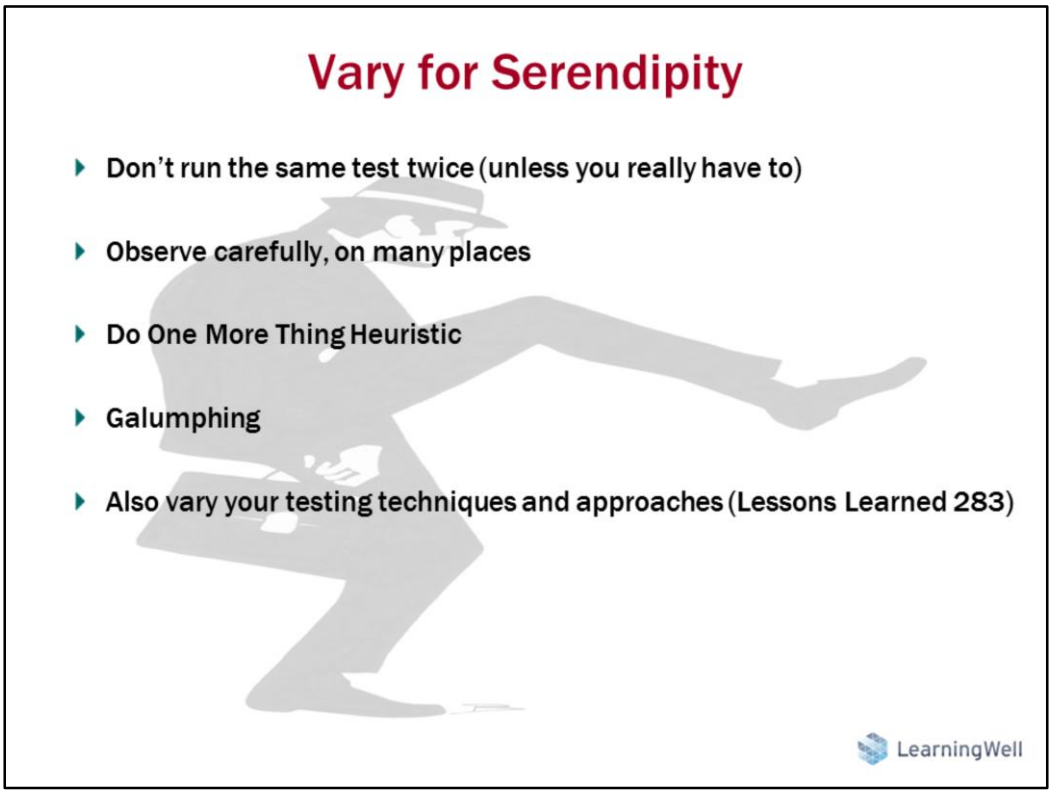
LearningWell

One part of serendipity opportunities lies in preparations. We know what computers we are using and which test data that is involved, and there are many ways to change these to make better chances for serendipitous findings.

One method for this I call the ”Error-Prone Machine”. On purpose, I don’t use the same settings as everyone else. Since the developers have English or Swedish settings, I use German or Japanese Regional Settings, so I have different date and decimal formats.
I have changed my temp folder, I have the task bar to the left of the screen, a friend of mine have never ever installed into the default location.
I use high DPI, and of course make sure to show script errors in the browser.
These changes don’t find problems often, but when they do, I get them for free.

I also often use the ”Background Complexity Heuristic”, where I use more complex data than necessary. So if I would test the search functionality in Word, many tests could be performed with a few sentences in the document. I would often prefer to have a very complex document, maybe with 200 pages, images, footnotes, formatting etcetera. This complexity is not needed, but it increases my chances of serendipity.

I guess you have or can come up with good preparations that make your tests a bit richer.
Many preparations just have to be done once, so it’s often a well worth investment.



Testing is about finding new information, so to run the same tests over and over again is a safe strategy to be blind for new and important things.
Of course, sometimes we want to do this, for instance certain regression tests that we want to know work every day.
But if you want to increase your coverage you should do new tests, or at least new variations. It might be to use keyboard instead of mouse, it might be to do things in different order, it might be to use new kinds of data as often as possible.

You can also vary how you look at your testing and the results. You can look at many places, on screen, in database, in code, in log files, after export etc.
The careful observation is a key, if we don't look with a curious mind, we won't see stuff.

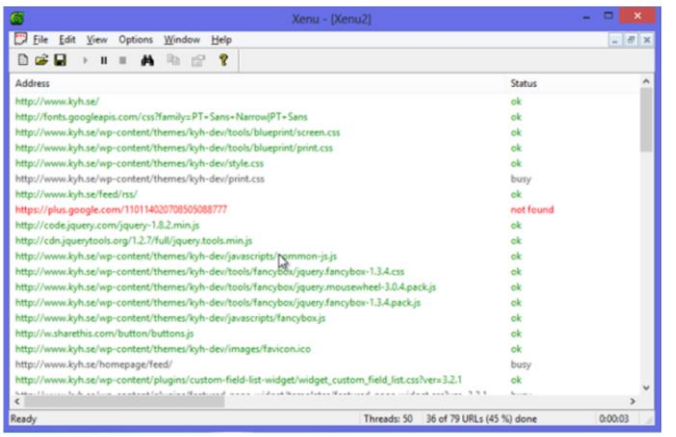
The Do One More Thing Heuristic is used after you have completed any test. Additionally do something error-prone, something popular or what a user might do. Don't think too much; just do something, and see what happens. It could be to copy data and paste in an e-mail, it could be to press F1 to read the Help, it might be do another action that you feel is worth trying. This is a way to add more complexity, and it is not for free, but almost.

James Bach has a lot of material on Galumphing, to do things in over-elaborated ways. This is not only because it is fun, it is because the variations will help you discover things about what you are testing. My most vivid example was a dialog that would crash when you clicked Cancel, but only if you first had moved the dialog box. It was not something I did on purpose, I didn't think "maybe there will be a crash on cancel if I first move the dialog?" No, it was a serendipitous finding because I uncounsciously added variations by doing things I didn't have to.

My favorite testing lesson comes from the book Lessons Learned in Software Testing, it's number 283:
"It is better to test pretty well in many ways, than perfect in one or two."
This is because important things often are missed because you looked at the software with too few approaches.
You should vary how you test, so if you do mostly free-form exploratory testing, maybe specification-based tests is your next best step.
This goes back to the potato, we don't want to look at just a small part of the potato, we want to learn more and more about it over time.

Serendipity Observation Examples

- **Look carefully**
- **Visualizing**
- **Notepad Heuristic and more**



LearningWell

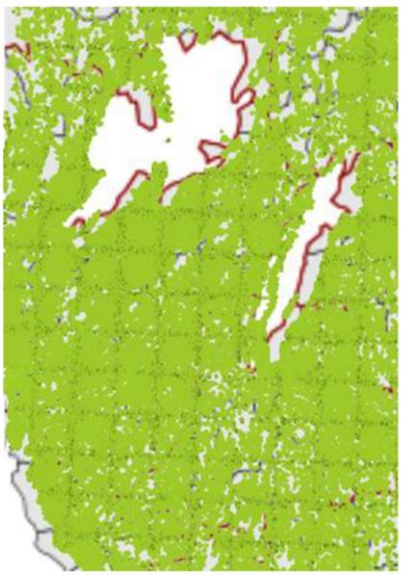
Now let's look at some examples of serendipity in testing.

When I taught testing tools at a school, I showed them the Xenu Link Checker. I used their school's web site as live example, as I like to test new things, to make it more for real (also for myself.)
As I browsed the list of results, especially looking at the "red", broken links, a student said:
"Wait a minute, what is that?"
So I stopped, and scrolled back on his request.

And what we looked at was green, valid links, but they linked to escortistanbul and similar.
Not very appropriate for a school, right?
Thsi is serendipity!

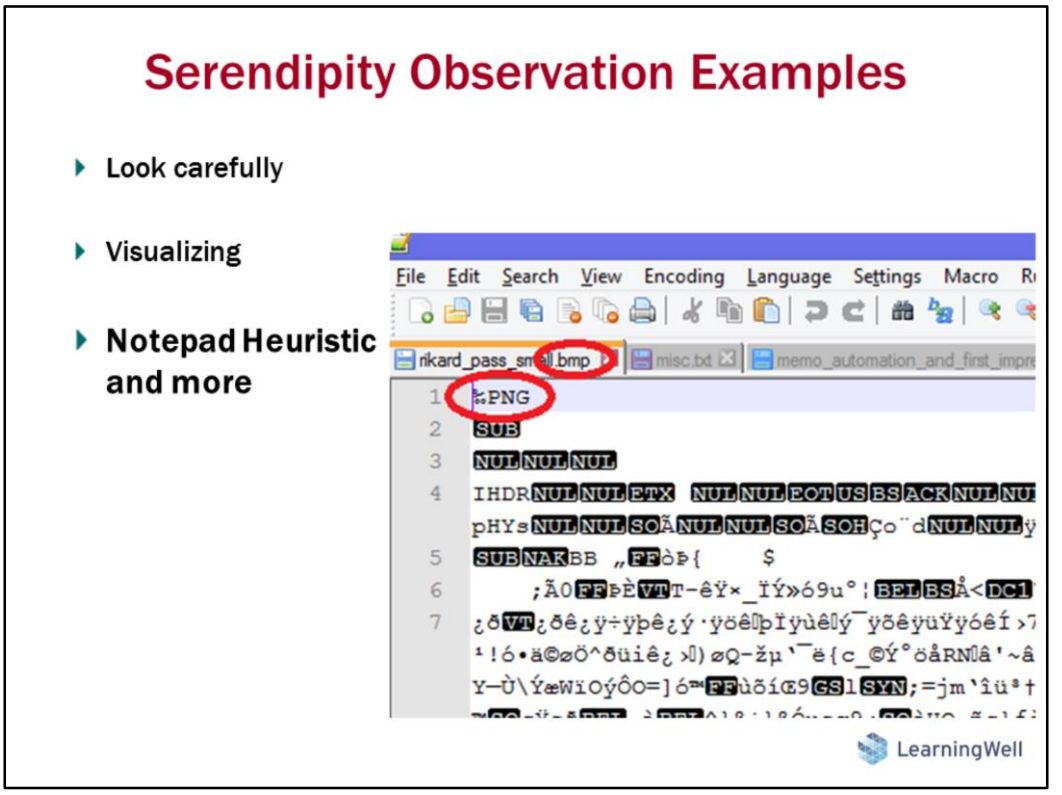
Serendipity Observation Examples

- Look carefully
- **Visualizing**
- Notepad Heuristic and more



LearningWell

My second example happened when we tested an application for fire departments and fire risk in Sweden. There were strange results near Uppsala, and by visualizing the whole underlying database we could see that there were holes in the data for some municipalities. But of course we looked some more at the data visually, and by doing some filtering, we noticed that there were unexpected patterns all over Sweden. The risk of fires were arranged according to a square-like net that you can see in the image, not at all emulating reality. Big risk of fire can't be arranged in this pattern, so we knew it was wrong, but not why.
All the underlying data had to be rebuilt.
It's serendipity!



Quite often I open any file in a text editor, it's the Notepad Heuristic. Even if it is a binary file, you can see interesting stuff, for example that a file created as a .bmp actually was a .png.
Now and then I look at log files for no particular reason, just to see if something interesting shows up.

Another recent story is when I did performance tests, and the guy with admin control thought the server had lost control and took it down.
This was a very good thing, because then we could see that the legal tracability would continue when restarted.
And it worked! But when looking carefully at the data logged, the timestamp was sending time, not viewing time, which is very wrong in this context.
I wish I could promise that this would have been covered anyway, but serendipity made me not having to find out.

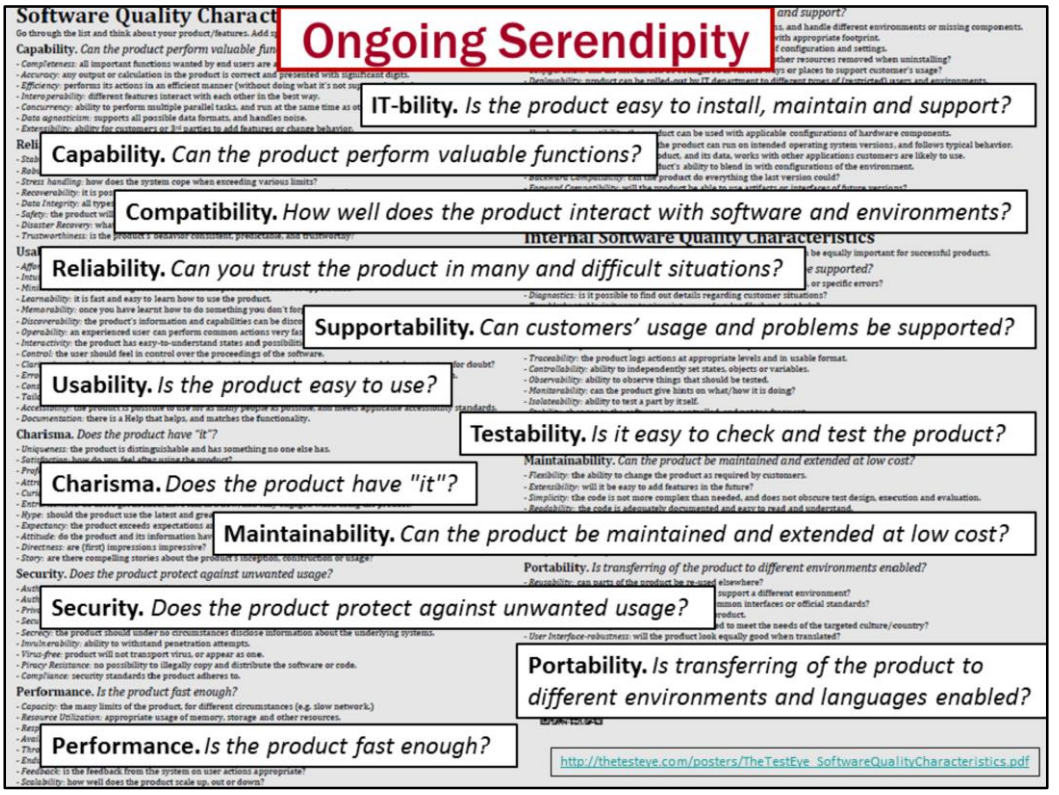
A pattern I see in my testing is that I perform many tests fast, with different kinds of data.
I never write out the details in advance, and I look for many things at once. It might seem unstructured (well, even to me…)
But I learn a lot about the software, and I can write test ideas afterwards.
I setup situations where serendipity can work to my advantage.

Ongoing Serendipity

- Things we know always matter, can be tested “for free”
- Make tests richer.

LearningWell

Serendipity is not something you ”do”, it is something that happens.
One way to make that happen more often is to have many ongoing test ideas for your product, things that you don’t need to think about, but when a violation occurs, you will notice.
Spelling errors is a typical example, and many others can be found by elaborating on quality characteristics.
Things we know always matter can be tested almost for free, and this will make your testing richer.



This long list is free to download, just google “Software Quality Characteristics”.

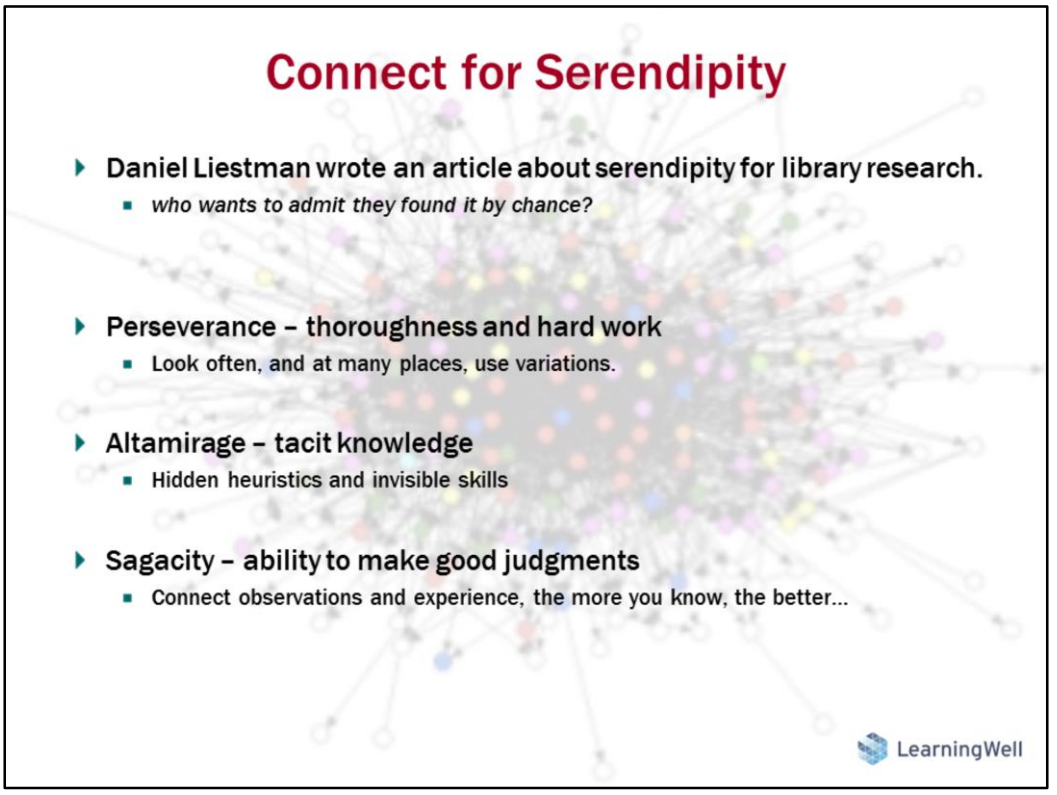
It is a thorough extension in the same spirit as James Bach’s Quality Criteria in the Heuristic Test Strategy Model.

If you figure out what Reliability and Usability really means in your situation, you can spot problems with that, regardless of what you are testing.

If you know your Charisma, you can spot a violation in a corner that few others will examine.

All of these characteristics have more details to them, and they mean different things for each unique product.

Find out the “hows” of your product, and you will get more serendipity in your daily testing.



Daniel Liestman wrote an article about serendipitous findings for library research, which has been very helpful for my material.
A key point he makes is “who wants to admit they found it by chance?”
This is valid for testers as well, and I think it is the reason why serendipity isn’t more widely talked about and accepted.
That’s why most testing techniques comes from computer books, that’s why we often create a lot of seemingly impressive documentation.
It’s not easy to say “well, we learn as much as possible, so when we see the software in action, we will stumble on the most important things.”

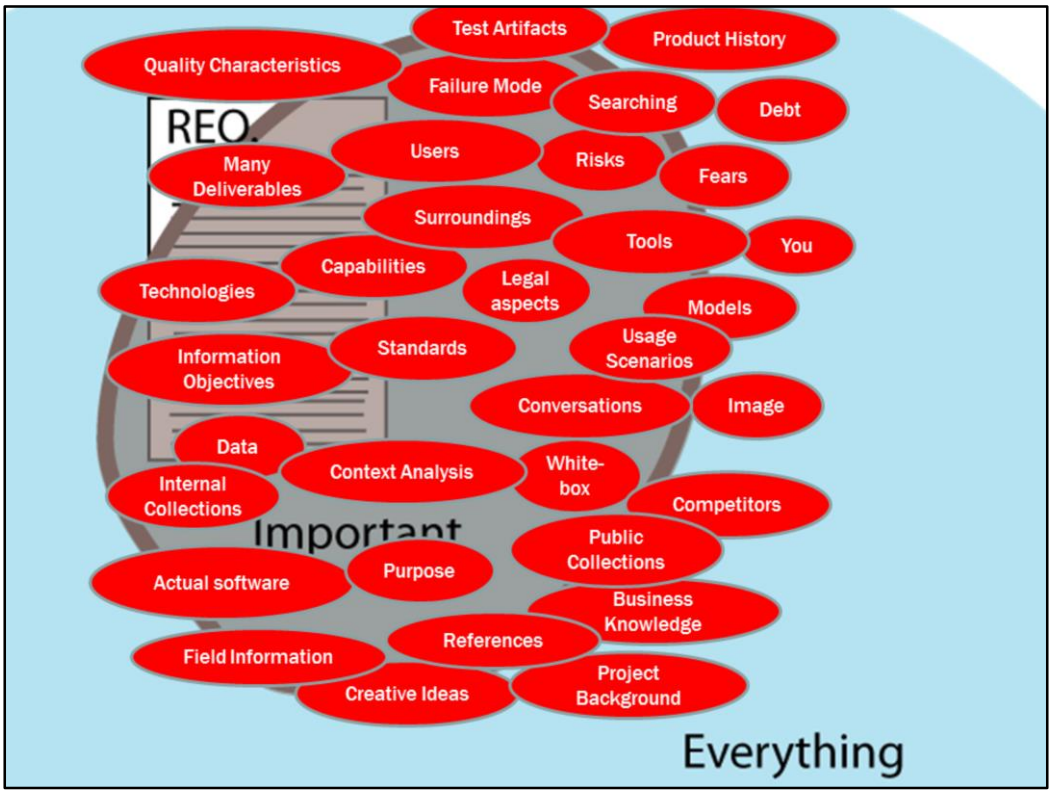
But if we don’t talk about it, we won’t get better at it.
If this helps us why should we hide it?

Liestman talks about perseverance, a tester that also does hard and boring work will have greater chances of serendipity.
Testing is fun, but not always, and the perseverance is needed to do many tests, also the boring ones, and eventually good things will happen (otherwise we might have a faulty test strategy?)

Altamirage is a rarely used word about your hidden heuristics and invisible skills. I see it as a part of testing’s tacit knowledge, things we know how to do, but can’t really explain. For example when we decide to continue the testing, but from another angle. We get new ideas as we see the software in action, and this grows from experience, but also from interacting with other testers, and discussing what you do.

It’s also about sagacity, the ability to make good decisions. The more you know, the better chances you have for this. Connect observations and experience, for instance when we notice an odd behavior we want to pursue.

Surrounding all of this is an understanding about what is important.
And saying ”learn a lot!” is quite fluffy, so let’s give this some more flesh.

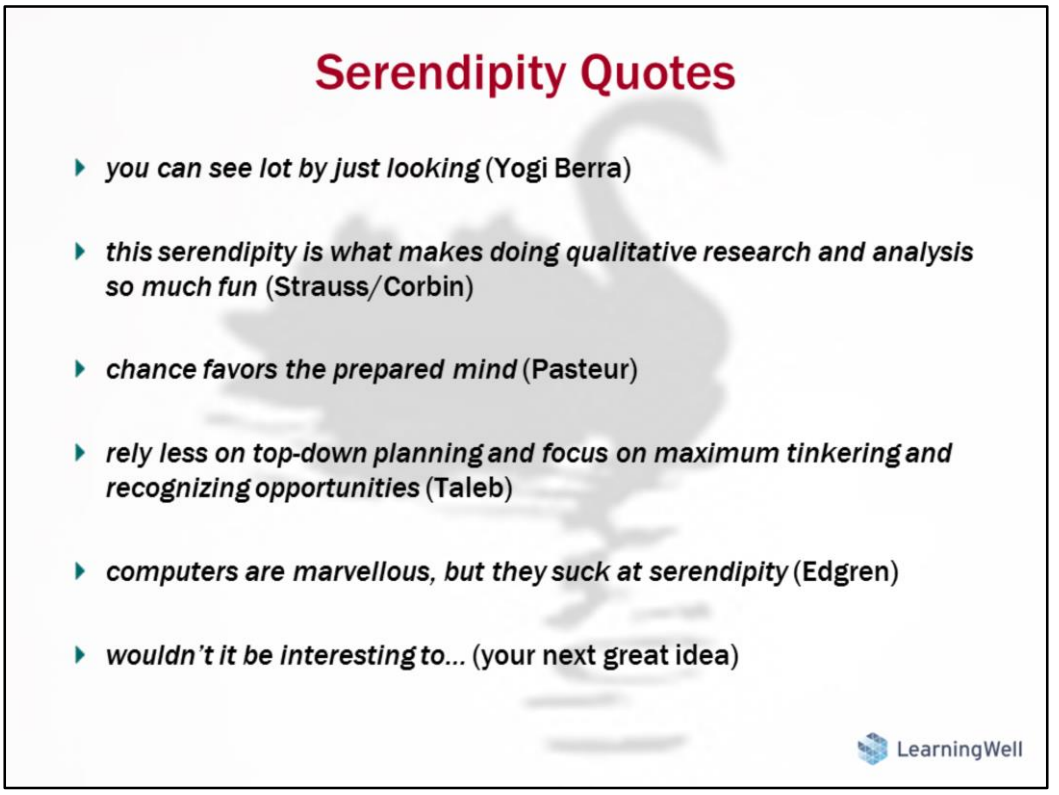


What we do is that we fill the potato with things like:

- Creating many models that enhance our understanding and testing ideas
- We explore the data in all its forms
- We learn about the underlying technologies and tools to explore them
- We learn about the business, so we understand the users' true purposes
- We have conversations with many people

…and all of this together builds up our understanding, and also enables good testing, with a lot of serendipity.

The more we know, the more value you get from just using the software.



Here are some serendipity quotes I like.

Yogi Berra is spot on: you can see a lot by just looking. And also the opposite: if you don't look, you won't see anything.

Strauss and Corbin are social scientists in the Grounded Theory tradition. They point to one of the reasons why I think testing is so fascinating. We have a lot of good ideas, but the unexpected serendipitous findings are really exciting!

Louis Pasteur puts emphasis on your prepared mind; your mind that knows a lot, and that also is ready to see new and interesting things. If you don't expect to see new things, you won't see them.

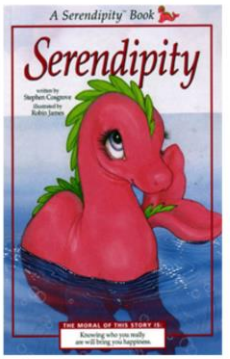
Taleb's quote about "maximum tinkering and recognizing opportunities" comes from the book The Black Swan, and is spot on to my message. Postive Black Swans happen, but you have to allow yourself tinkering and playing around, and you need to have the ability to recognize opportunities when they show up.

And I love computers and tools, but we need to know their strengths, and beware of the weak spot that has implications for software testing; humans can do more. Computers are marvellous, but they suck at serendipity.

To stimulate your thinking, say to your colleagues: "Wouldn't it be interesting to…." and then you force yourself to suggest a way of testing that enables serendipity.

Serendipity Summary

- Software testing is oozing with serendipity.
- Serendipity can be your friend and rescue, don't hide it.
- Learn a lot, prepare, do many tests and observe!



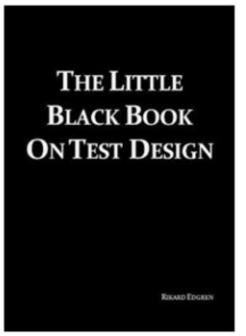
LearningWell

I hope you have gotten some tips on how to embrace serendipity in your daily practice.
To me, and all testers I know, serendipity is quite common, and it is working to our advantage.

I will end this with some fluffy words that you need to work out details for yourself: Learn a lot, prepare, do many tests and observe!

References

- Further reading:
  - Liestman, Chance in the Midst of Design: Approaches to Library Research Serendipity
  - Edgren/Emilsson/Jansson, Software Quality Characteristics
    http://thetesteye.com/posters/TheTesteEye_SoftwareQualityCharacteristics.pdf
  - Edgren/Emilsson/Jansson, 37 Sources for Test Ideas
    http://thetesteye.com/posters/TheTestEye_SourcesForTestIdeas.pdf
  - Edgren: The Little Black Book on Test Design
    http://thetesteye.com/papers/TheLittleBlackBookOnTestDesign.pdf



www.thetesteye.com rikard.edgren@thetesteye.com

LearningWell

So this was my talk, and after Daragh takes over, I will try to answer your questions (and maybe I will learn important things I didn't know I was looking for!)

Thanks for listening.