

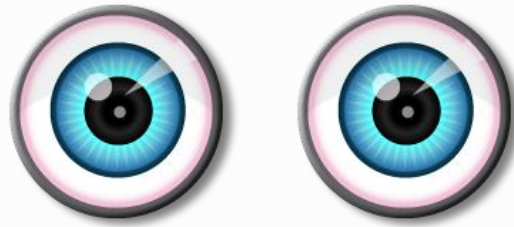
# 5 Steg till Kreativ Testning

Rikard Edgren  
TIBCO Spotfire  
SAST 15 år 2010-10-15

# Agenda

- Vikten av kreativitet
- En miljö för kreativitet
- ”Fuska” med kreativiteten

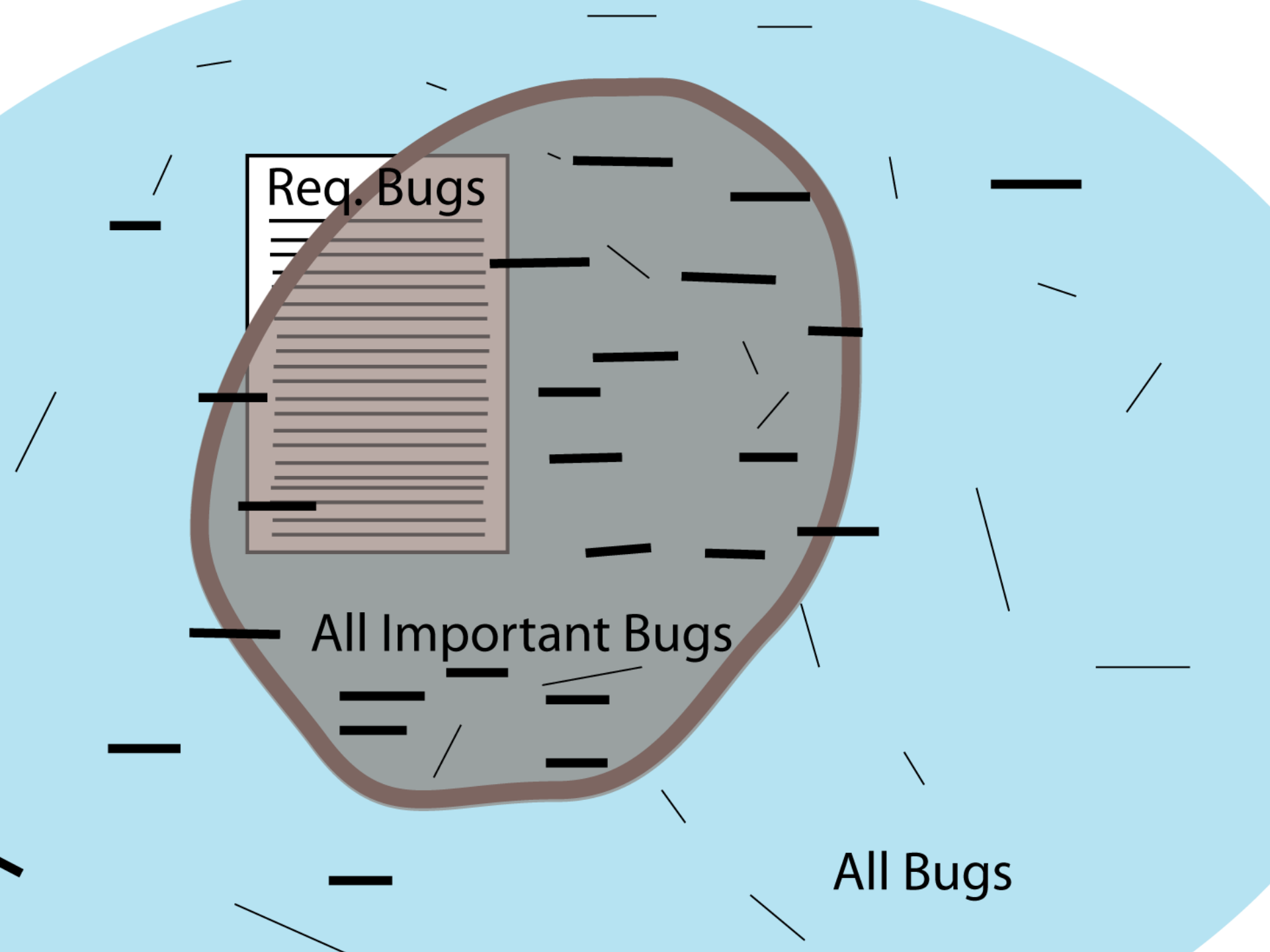
# Kreativitet



- Var skulle du vilja ha ett extra par ögon?

# Kreativitet & Testning

- ”Processen att få nya idéer som har värde”
  - stämmer oerhört ofta för testning!
- Den dagliga kreativiteten
  - identifiera tester
  - exekvera tester
  - utforska märkliga beteenden

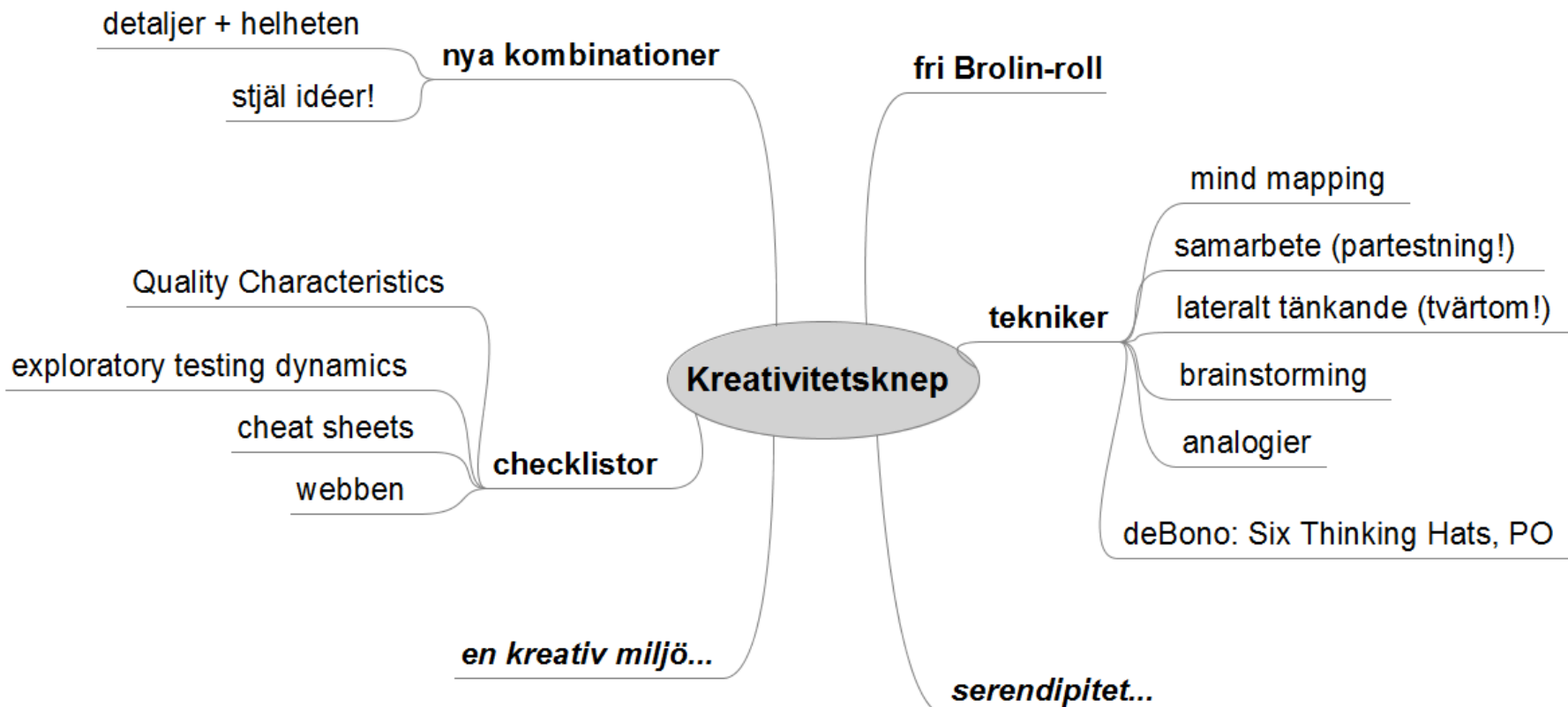


# En Kreativ Miljö

- kompetens
- mångfald
- tillit
- tolerans
- humor\*
- disciplin\*
- generositet
- gemenskap
- nyfikenhet
- frihetsanda
- *småskalighet*
- jämlikhet

\* inte **Sahlin**: *Kreativitetens filosofi*

# Kreativitetsknep



# Finn Fem Fel



...fast du har inget facit, och vet inte hur många fel som finns  
...och det kan finnas andra saker som är mycket viktigare



# Kreativ Testning



- Testögat
  - vill se fel
  - ser många sorters fel
  - tittar på många ställen
  - tittar ofta
  - fokuserar på det som är viktigt

# Serendipitet

- “man letar efter något, men hittar något helt annat, som är värdefullt”
- extremt vanligt för testare med frihet
- tillfälligheter + skarpsinne
- kan vara räddningen (testning är aldrig heltäckande)

# Sammanfattning

- kreativiteten växer framför allt där den tillåts växa
- hitta metoderna som passar dig
- *4 långa, men roliga steg att ta!*

# Frågor

- ???

[www.thetesteye.com](http://www.thetesteye.com)

[redgren@tibco.com](mailto:redgren@tibco.com)

# SOFTWARE TESTING INSPIRATION

thetesteye.com v0.6

## MODELS

**Requirements:** explicit, combinatorial, implicit, incorrect, changing, vague, different, impossible, aiding  
**Specifications:** conceptual, architectural, technical, functional, design, test specifications  
**Code:** old, new, shaky, read, reviewed?  
**Documentation:** online help, pdf, web sites  
**Actual software:** prototypes, in progress, ~~released~~, competitors  
**Other:** people's models, visions, images, **the whole picture**

## PROJECT

**Risky functionality:** recently "improved", core, problematic, difficult, high interoperability, complex, popular  
**Risks:** important, omitted, forgotten, changing, unknown  
**Project plan:** when, what, how?  
**Process:** Agile, Waterfall, home-brew, collaborative?  
**Infrastructure:** configuration management, test environment  
**Test execution context:** what, when, where, why, who, how?  
**Quality objectives:** what is always important?  
**Information objectives:** what information are stakeholders interested in?  
**Deliverables:** executables, interfaces, all sorts of documentation, Release Notes, readme:s, meta data  
**Unknown:** there always are unexpected changes and inspiring situations

## TECHNOLOGY

**Infrastructure:** hardware, OS, network settings, co-existing applications, 3<sup>rd</sup> Party Components  
**Customer environment:** Are your lab condition close to the ones in the field?  
**Tools:** development tools, (static) test tools, monitors, editors, brain  
**Systems:** what does the small and big system look like?  
**Future:** what about the forthcoming technology?

## PEOPLE

**Team:** developers, interaction designers, leaders, managers, owners, technical writers, testers, experts  
**(Hidden) Stakeholders:** obvious and non-obvious interests and investments  
**YOU:** your knowledge, experience, subjectivity and skills (analytical thinking, critical thinking, creativity, test factoring, investigation, problem solving, The Test Eye)  
**Users:** needs, knowledge, feelings, impairments, personas, have you seen the real users in action?

## USAGE

**Support department:** what experiences are channeled to support?  
**User scenarios/Use cases:** do you know many ways of actual usage?  
**Customer stories:** what problems are the customers trying to solve?  
**Dog food:** can you use the product internally "for real"?  
**Competitors:** software products, in-house tools, analog systems  
**Training material:** learn about how customers learn your software  
**Business:** needs, logic, information, knowledge, standards, laws

## HISTORY

**Previous versions:** artefacts, right/wrong behavior?  
**Bugs/Error catalogs:** what bugs occurred for similar functionality?  
**Test Ideas, Cases, Strategies, Results:** what can you learn from previous test efforts?  
**Claims:** which features are used in marketing material?  
**Reviews:** what has been said by others about your product?

## SOFTWARE TESTING

**Quality Characteristics:** Usability, Reliability, Usability, Charisma, Security, Performance, IT-bility, Compatibility, Supportability, Testability, Maintainability, Portability, Localizability  
**General test ideas:** quicktests, tours, mnemonics, heuristics, check lists, bug taxonomies  
**Tricks:** attacks, Cheat Sheets, error-prone machine, Basic Configuration Matrix  
**Information:** books, articles, courses, blogs, forums, web sites, conferences, conversations  
**Testing theory:** with many different techniques/approaches you have a higher chance of finding important information