



Ohjelmistotestauksen opetuksen kokemuksia fantasiatarinan avulla

Kari Kakkonen,
Dragons Out Oy

ITK 2022

6.10.2022

Kari Kakkonen



twitter.com/kkakkonen
linkedin.com/in/karikakkonen/
Dragonsout.com

• ROLES

- Knowit Solutions Oy, Director of Training and Competences, Lead Consultant, Trainer and Coach
- Children's and testing author at Dragons Out Oy
- TMMi, Board of Directors
- Treasurer of Finnish Software Testing Board (FiSTB)

• ACHIEVEMENTS

- **Tester of the Year in Finland 2021**
- **EuroSTAR European Testing Excellence Award 2021**
- ISTQB Executive Committee 2015-2021
- Influencing testing since 1996
- Ranked in 100 most influential IT persons in Finland (Tivi magazine)
- Great number of presentations in Finnish and international conferences
- TestausOSY/FAST founding member.
- Co-author of Agile Testing Foundations book
- Regular blogger in Tivi-magazine

• EDUCATION

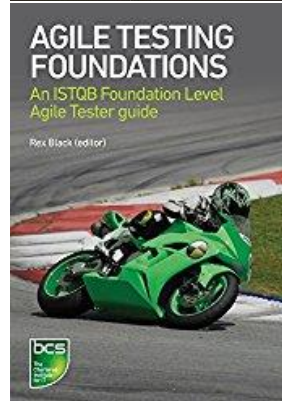
- ISTQB Expert Level Test Management & Advanced Full & Agile Tester certified
- DASA DevOps, Scrum Master and SAFe certified
- SPICE provisionary assessor certified
- M.Sc.(Eng), Helsinki University of Technology (present Aalto University), Otaniemi, Espoo
- Marketing studies, University of Wisconsin-Madison, the USA.

BUSINESS DOMAINS

- Wide spread of business domain knowledge
 - Embedded, Industry, Public,
 - Training, Telecom, Commerce,
 - Insurance, Banking, Pension

SERVICES

- ISTQB Advanced, Foundation and Agile Testing
- A4Q AI and Software Testing
- Knowit Quality Professional
- DASA DevOps
- Quality & Test process and organization development, Metrics
- Agile testing, Scrum, Kanban, Lean
- Leadership
- Test automation, Mobile, Cloud, DevOps, AI
- Quality, Cost, Benefits.





Creating a Nordic powerhouse for digital solutions with a sustainable impact

knowit



**CYBERCOM
GROUP** 

3,800+
/ Experts

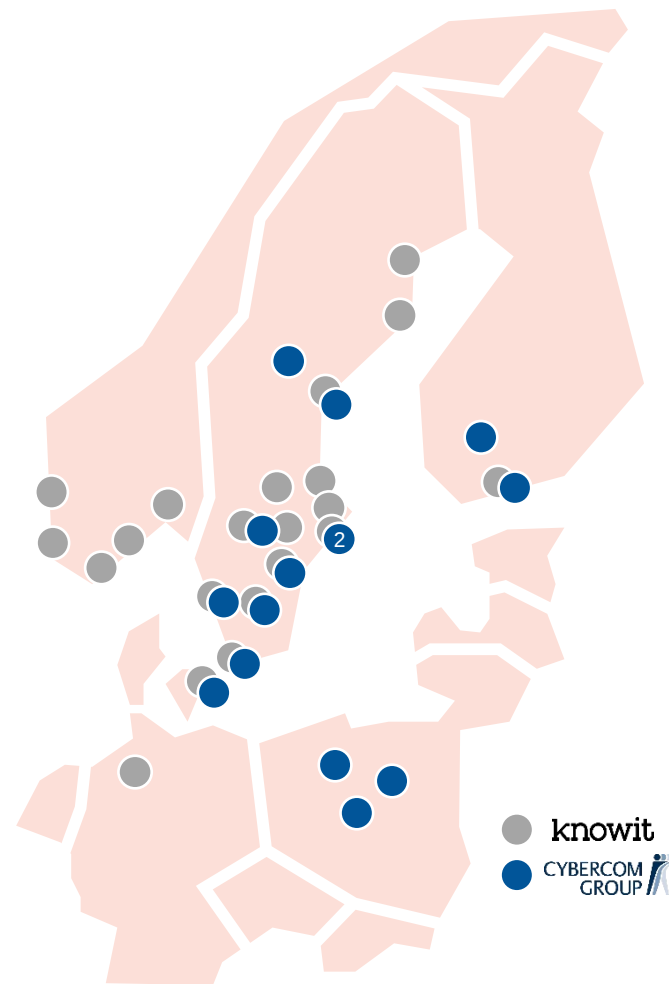
4 Business Areas
/ Solutions, Experience,
Insight ja Connectivity

545 M €
/ Liikevaihto

6 Countries
/ Sweden, Norway, Finland, Denmark,
Germany and Poland

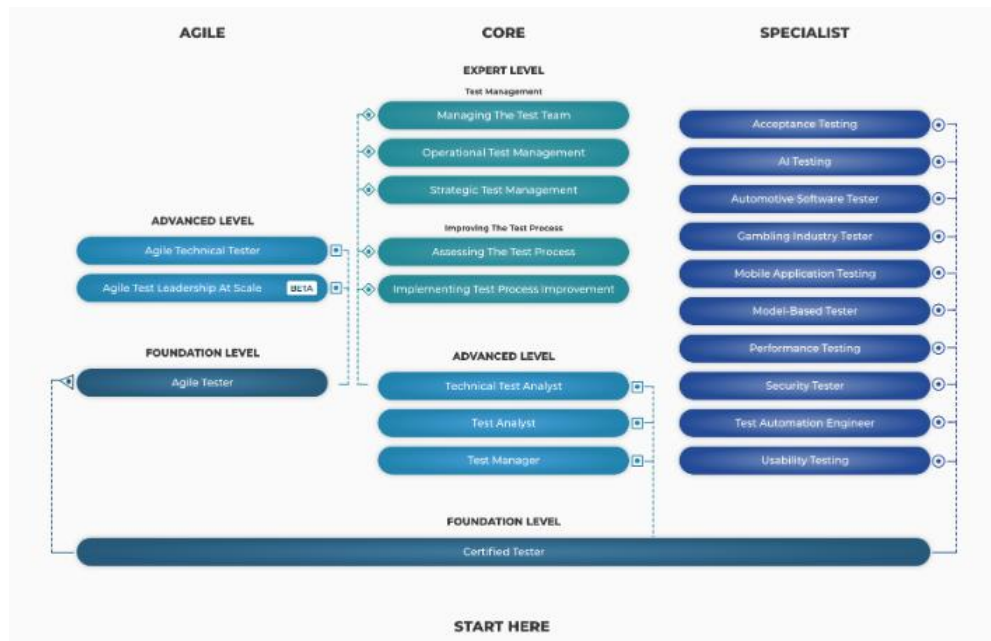
Nordic ESG champions
/ Clear vision to accelerate
the UN Environment, Social,
Governance, and
Sustainable Development
agenda

54,3 M €
/ EBITA



ISTQB GLOBAL PRESENCE

- Number of exams administered: **1 030,000**
- Number of certifications issued: **750,000**
- In **129 countries**



Countries covered by Member Boards and Global Exam Providers

Countries covered by Global Exam Providers

TMMi for test improvement in all kinds of testing, including agile and DevOps



Agenda

- Miksi testaustakin pitää oppia koodauksen rinnalla?
- Kirjaprojekti
- Miten fantasiatarina innostaa sopivana vertauksena koodauksen ja ohjelmistotestauksen maailmaan?
- Mille oppitunneille testausopetus sopii?
- Q&A



Miksi testausta pitää oppia koodauksen rinnalla?

Miksi testausta?

- Koodaus tarvitsee aina rinnalleen testausta
- Testaus on muutakin kuin koodaajan omat kokeilut
- Testaus on jopa yli puolet kaikesta ohjelmistokehitystyöstä
- Testaus lisää
 - Ohjelmisto laatua
 - Säästää käyttäjien hermoja
 - Parantaa yritysten kykyä toimia
 - Säästää mailman resursseja

Miksi testausta lapsille?

- Tieto- ja viestintätekniikan taidot ovat osa kansallista opetussuunnitelmaa
 - Käytön taidot
 - Koodaus
- Koodaukseen on oppimateriaaleja, kirjoja, kerhoja
- Ohjelmistotestaus puuttuu usein koodausopetuksen kyljestä täysin
- Oppimalla pelkästään koodaamaan syntyy massiivinen laatuvelka
- Testauksen oppiminen on kestävän ohjelmistokehityksen oppimista
- Testaus on myös oma ammattikuntansa – koodaus ei ole läheskään ainoa tehtävä Tieto- ja Viestintätekniikan alalla



Kirjaprojektin tiivistelmä

Tahdon kertoa tarinan



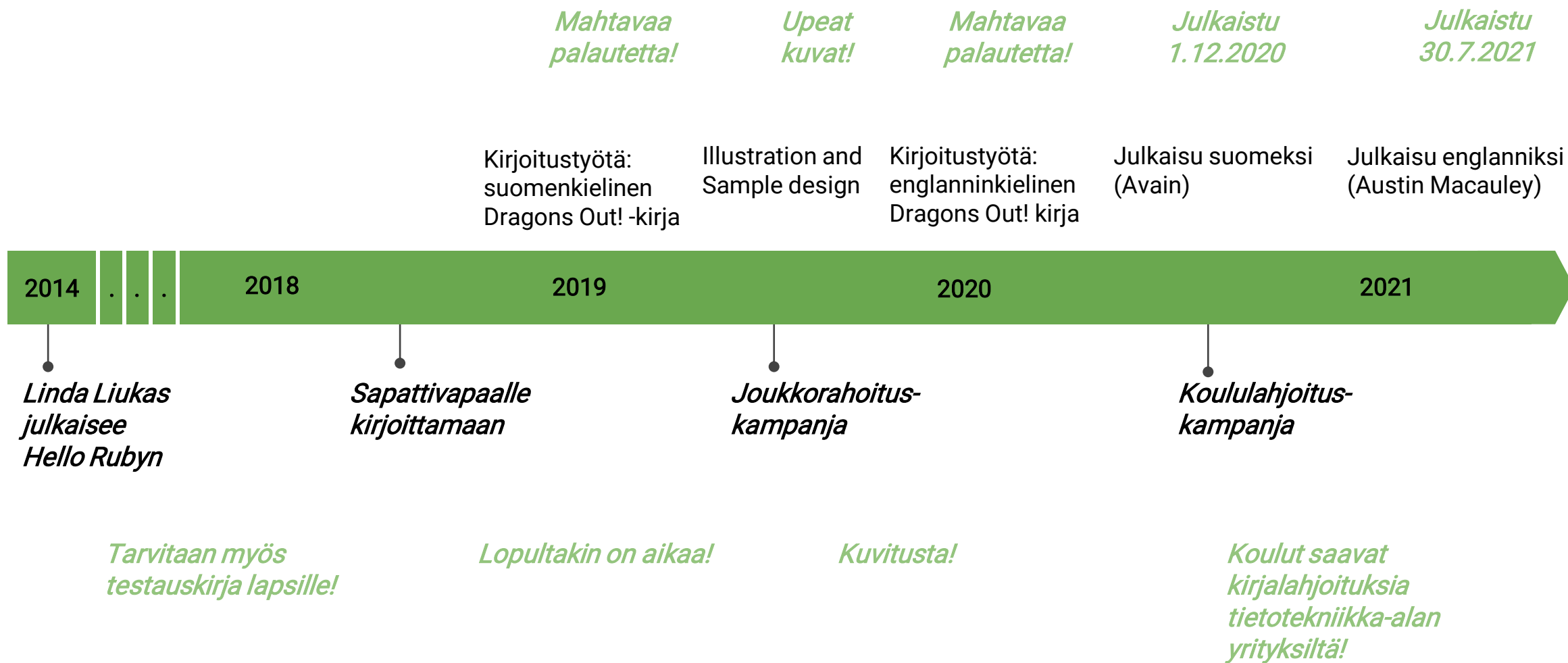
“Jokaisella on tarina. Jokainen asia tarvitsee tarinankertojan. Opi olemaan tarinankertoja, sillä jollet ole tosi-tv:n esiintyjä, kukaan ei kerro tarinaasi sinun puolestasi.

Joten kerro meille tarina. Kerro hyvä tarina. Ja anna tuon hyvän tarinan tulla osaksi tarinoiden sinfoniaa, joka tekee tästä maailmasta paremman paikan.

James Whittaker

<https://medium.com/@docjamesw/the-storytelling-manifesto-f17548a358b3>

Kirjaprojektin pääkohdat



Kirja ”Dragons Out!”

- Kirjailija Kari Kakkonen
- Kuvittaja Adrienn Széll
- Oikeudet kuviin ja tekstiin Dragons Out Oy
- Tästä esityksestä on opettajille (tai kelle vain) maksuton Creative Commons – lisensoitu versio kirjan verkkosivuilla
 - Käännetty monille kielille!
- Lisätietoja: www.dragonsout.com



Miten fantasiatarina innostaa sopivana vertauksena
koodauksen ja ohjelmistotestauksen maailmaan?

Tarinan voima...

Tarina

- Joutsenjärvi käänsi hevosensa ympäri ja ratsasti nopeasti takaisin paaluvarustukselle. Hän huikkasi apunaan olleille ritareille ja rakennusmestari Vahva-Einolle, että lohikäärme oli tulossa. Kaikki teroitettut hirret pitäisi kasata äkkiä paaluvarustuksen aukolle. Keihäät ja miekat, kellä sellaisia oli, pitäisi hakea välittömästi. Vettä pitäisi varata ämpäreihin mahdollisimman paljon. Sitten hän meni etsimään Keltapartaa linnalta.

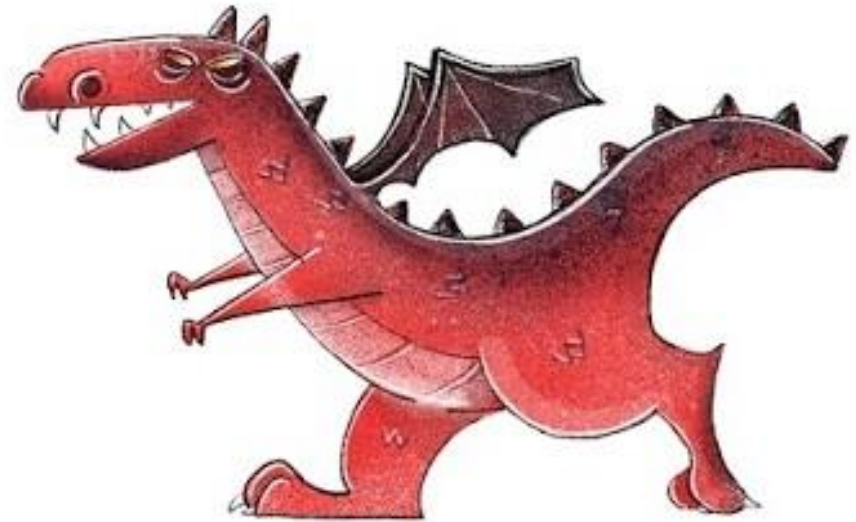
Selitys

- Tarinassa lohikäärme saapui kylään kesken paaluvarustuksen kunnostamisen. Vastaavasti suurin osa vioista ohjelmistoissa löytyy kehityksen aikana ennen kuin ohjelmisto on valmis. Silloin **vikojen löytäjät (testaajat) ja korjaajat (koodaajat)**, ovat koko ajan saatavilla. Yleensä testaaja löytää vian, eikä siis odota, että ohjelmiston käyttäjä löytäisi vian joskus myöhemmin. Tässä tarinassa Joutsenjärvi oli testaaja, joka löysi ja tunnisti vian eli lohikäärmeen. Hän ei pystynyt testaajana tällä kertaa itse korjaamaan tätä vikaa, vaan tarvitsi siihen myös koodaajia eli kehittäjiä.*



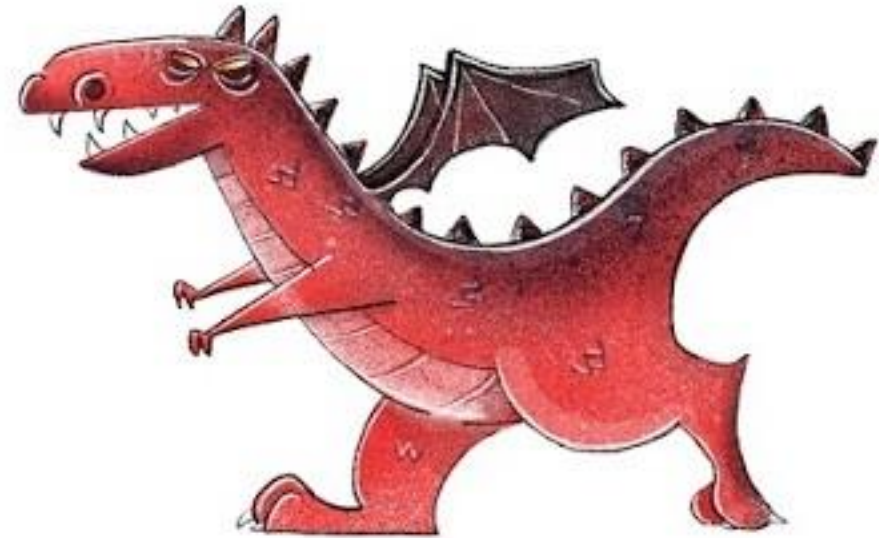
Ärsyttävä lohikäärme

- **Väri:** Punainen
- **Koko:** Keskitaso
- **Vaikeus löytää:** Vaikea
- **Vaikeus karkottaa:** Helppo
- **Lentää?:** Ei
- **Siivet:** Pienet
- **Syöksee tulta?:** Kyllä
- **Lempipuuha:** Lampaiden syönti



Ärsyttävä lohikäärme

- **Vian nimi:** Muistivuoto
- **Vakavuus:** Keskitaso
- **Vian oire:** Laite hidastuu, kunnes se ei voi enää tehdä mitään ja se sammuu
- **Vian syy:** Muistia varataan ohjelmiston käyttöön, mutta käytön jälkeen sitä ei vapauteta
- **Alkuperäissyitä:** Koodaaja ei ole huolellinen muistinvapauttamisessa. Ehkä ei osaa, ehkä ei muista.
- **Testaus:** Mitataan käytetyn muistin määrää samaan aikaan kuin ohjelmistoa käytetään. Jos käytetyn muistin määrä lisääntyy koko ajan, kyseessä on luultavasti muistivuoto.
- **Korjaus:** Ohjelmistoa ajetaan koodirivi kerrallaan, kunnes löydetään kohta, jonka voi korjata. Muisti vapautetaan sopivalla koodilla.



Ilkeä lohikäärme

- **Väri:** Musta
- **Koko:** Pieni
- **Vaikeus löytää:** Vaikea
- **Vaikeus karkottaa:** Keskitaso
- **Lentää?:** Kyllä
- **Siivet:** Keskitaso
- **Syöksee tulta?:** Paljon
- **Lempipuuha:** Ruoan ja aarteiden ryöstäminen salassa



Ilkeä lohikäärme

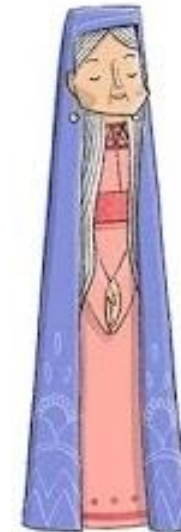
- **Vian nimi:** Tietoturvakäpälä
- **Vakavuus:** Suuri
- **Vian oire:** Ohjelmiston säilyttämää tärkeää tietoa löytyy muualta (vaikkapa pankkikorttitiedot). Voi olla myös, että ohjelmisto ei toimi tai toimii väärin.
- **Vian syy:** Rikollinen on käyttänyt tietoturvakäpälää ja murtautunut järjestelmään, sitten varastanut tai tuhonnut jotain.
- **Alkuperäissyitä:** Koodaaja ei ole noudattanut uusimpia tietoturvallisen koodauksen ohjeita. Ehkä ei osaa tai tiedä.
- **Testaus:** Etsitään ohjelmistosta tunnettuja tietoturvakäpäläitä käyttämällä ohjelmistoa itse tai tietoturvaohjelman avulla. Voi myös tutkia koodia. Apuna lista tunnettuja käpäläitä.
- **Korjaus:** Tunnettuun käpälään liittyy myös tunnettu korjaus. Se tehdään koodin, tai järjestelmän asetuksiin.



Käyttäjät

- Testausta voivat tehdä kaikki, mutta koodaajien ja testaajien odotetaan testaavan eniten.
- Ohjelmiston käyttäjät voivat osallistua testaukseen. Fantasiatarinassa kyläläiset, niin lapset kuin aikuisetkin, löytävät lohikäärmeitä.
- Tällainen testaus on usein hyväksymistestausta.
- Käyttäjät voivat myös auttaa ohjelmiston rakentamisessa.

Testaaja = ihminen, joka testaa. Ihminen, jolla on testaustaitoja. Voi olla testaaja, koodaaja, ylläpitäjä tai käyttäjä.



Ylläpitäjät

- Ylläpitäjät tarkkailevat ohjelmistoa ja pitävät sen käynnissä.
- Tämän vuoksi he sekä testaavat että poistavat vikoja.
- He myös auttavat käyttäjiä
- Ylläpitotiimi yrittää monesti pärjätä itseksensä ongelmien kanssa, mutta pyytää tarvittaessa apua kehitystiimiltä, kuten metsästäjäkin voi pyytää ritareilta apua lohikäärmeen tappamisessa.
- Joskus mukana kehitystiimissä (ohjelmistotiimi onkin DevOps-tiimi)

Ylläpitotiimi = ylläpitäjät, eli ihmiset, jotka huolehtivat ohjelmiston toimivuudesta, kun sitä käytetään.



DevOps – jatkuva kehitys

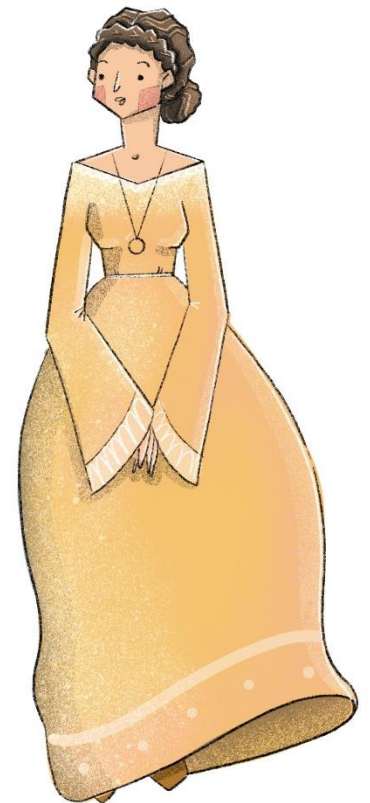
- Joskus ylläpitäjä on mukana kehitystiimissä, jolloin kehitystiimi onkin DevOps-tiimi, kuten tarinan metsästäjät välillä toimia yhdessä ritarien kanssa.
- Tällainen tiimi rakentaa ja testaa ohjelmistoa koko ajan, vie uusia ominaisuuksia käyttäjille ja samalla tukee käyttäjiä ohjelmiston käyttämisessä.

DevOps = ohjelmistokehityksen ja ylläpidon yhdistäminen.
Sama tiimi rakentaa ohjelmistoa ja ylläpitää sitä.



Ohjelmistojen tilaajat ja omistajat

- Tilaajat ostavat ohjelmistoja ja järjestelmiä kehitystiimeiltä.
- He voivat olla tuote-omistajia tai yritysten johtoa.
- He määrittelevät, mitä järjestelmissä pitää olla, mutta kuuntelevat kehitystiimiä, kuten tarinoissa linnanherrat ja linnanrouvat rakennuttavat linnoja yhdessä ritareiden kanssa.



Tilaaja = ihminen, joka pyytää rakentamaan ohjelmistotuotteen, eli ohjelmiston. Voi olla myös yritys, jota tietysti edustaa joku ihminen.

Asiantuntijat auttavat

- Kehitystiimit eivät osaa kaikkea hyvin, vaan tarvitsevat tukea erityisalueiden asiantuntijoilta, kuten fantasiatarinoissa erilaiset tietäjät auttavat kyläläisiä ja ritareita.
- Tyypillisiä asiantuntijoita ovat käytettävyyden, tietoturvan ja suorituskyvyn erikoisosaajat, jotka auttavat kehitystiimiä.
- Esimerkiksi käytettävyydasiantuntija monesti tekee tai ohjaa käytettävyydestausta.

Käytettävyydasiantuntija = ihminen, joka on erikoistunut tietojärjestelmien hyvän käytettävyyden suunnitteluun.





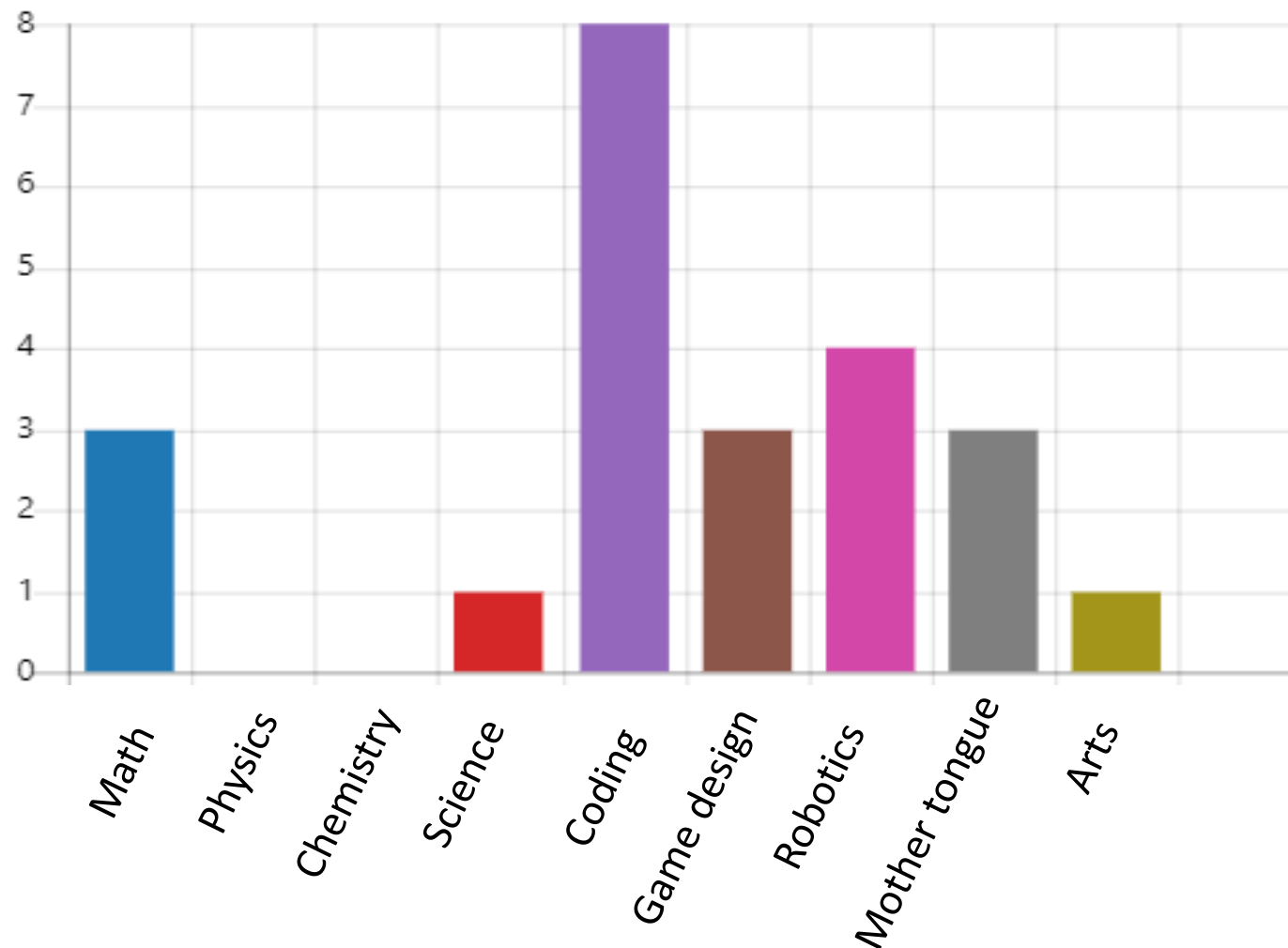
Mille oppitunneille testausopetus sopii?

Opettajien näkemyksiä ohjelmistotestauksen opetuksesta kysytty

- TVT-alan yritykset ja yhdistykset lahjoittivat 2020-2021 Dragons Out -ohjelmistotestauskirjoja 60 kouluun Suomessa
- Lahjoitettuja kirjoja vastaanottaneet koulut vastasivat kyselyyn
- ITK-messuilla 2021 osallistujat antoivat vastauksia samoihin kysymyksiin

Ohjelmistotestaus sopii monien aineiden tunneille

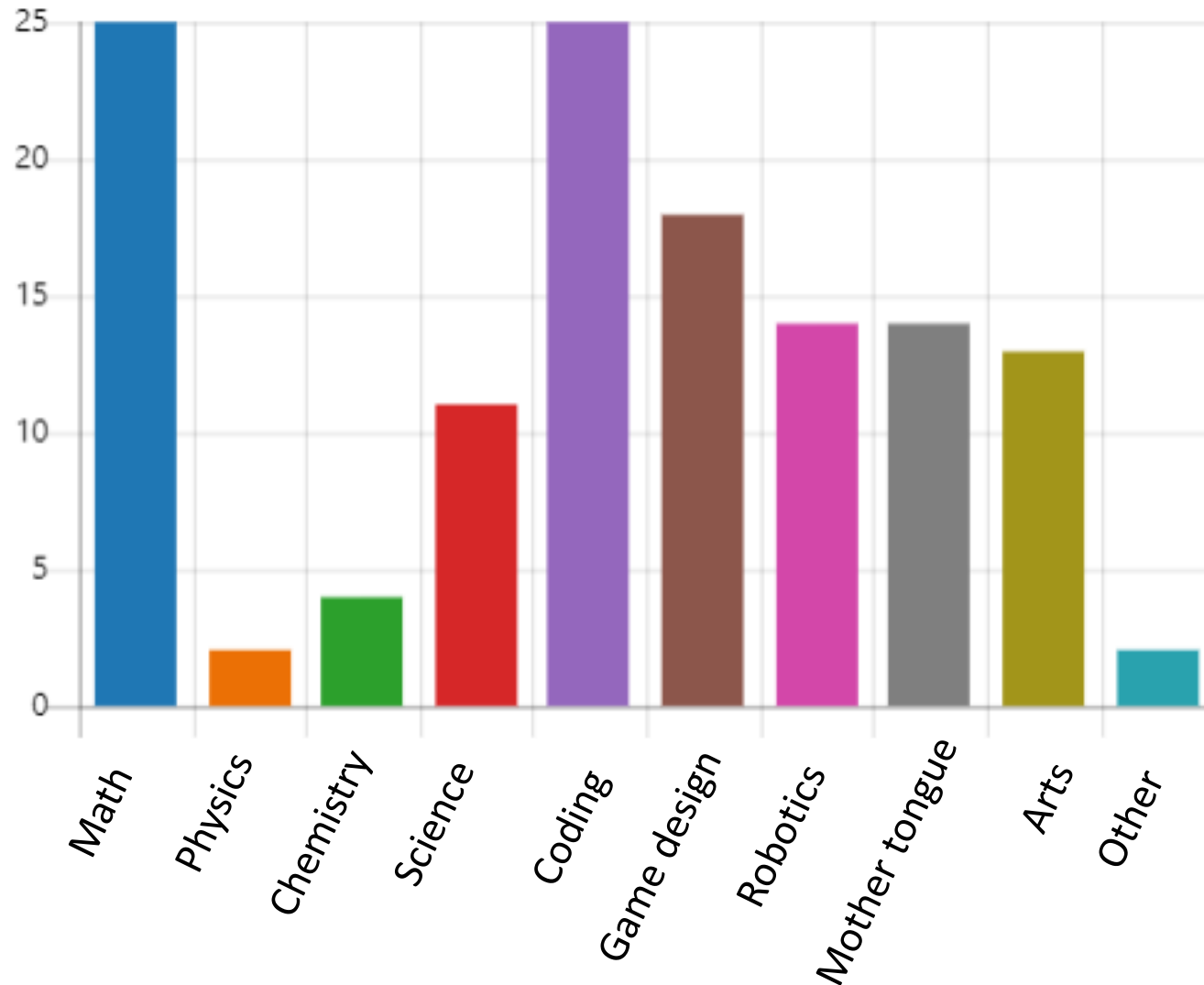
- Testaus sisältyy helposti moneen aiheeseen
 - Ilmiöoppiminen
- Koodaustunnit (luonnollisesti) sopii parhaiten myös testauksen opettamiseen



N=8 from teachers who used donated books

Matematiikka ja koodaus nähdään ohjelmistotestauksen kaverireina

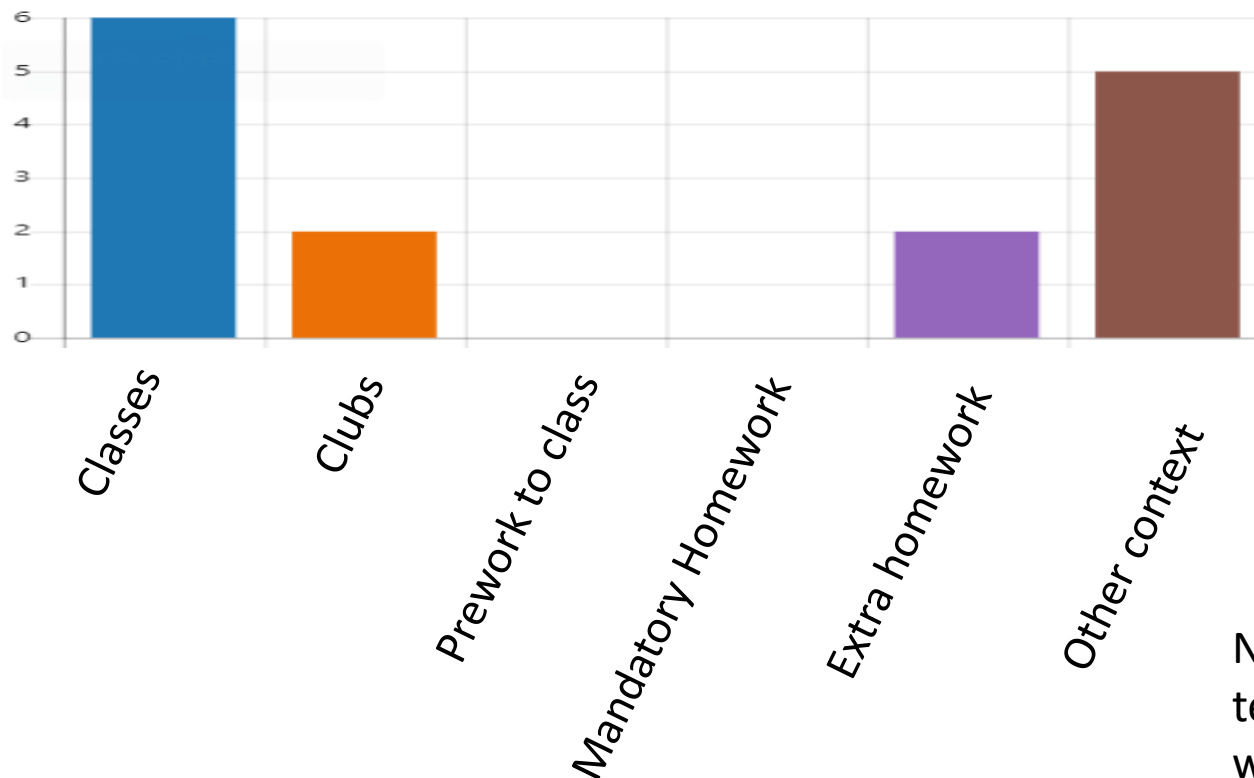
- Testaus sisältyy helposti moneen aiheeseen
 - Ilmiöoppiminen
- Koodaustunnit (luonnollisesti) sopii parhaiten myös testauksen opettamiseen



N=31
from
teachers
at trade
fair

Missä tilanteissa on käytännöllisintä oppia ohjelmistotestauskirjan avulla?

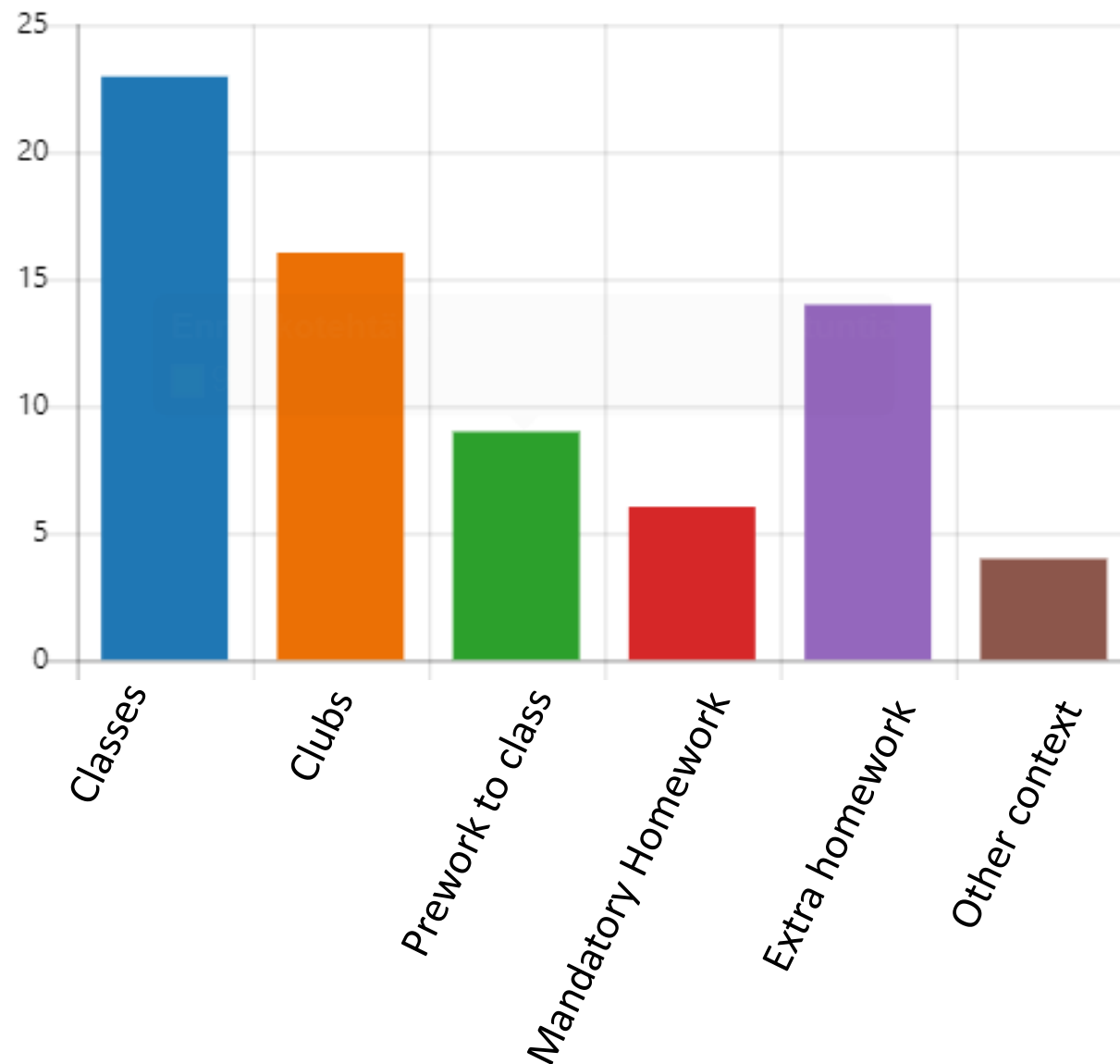
- Opettajan avustuksella oppiminen on käytännöllisintä joskin oppituntien ulkopuolellakin voi oppia
- Organisoitu oppiminen on parasta!



N=8 from teachers who used donated books

Missä tilanteissa on käytännöllisintä oppia kirjan avulla?

- Opettajan avustuksella oppiminen on käytännöllisintä joskin oppituntien ulkopuolellakin voi oppia
- Organisoitu oppiminen on parasta!



Oppimistyylien yhdistäminen

- Oppimistyylien käyttäminen yhdessä saa oppilaat innostumaan ohjelmistotestauksesta
- Yleensä yhdistetään
 - **Piirtämisharjoitukset** (piirrä lohikäärme / vika)
 - **Kuuntelemista** tai **lukemista** käyttäen testauskirjan sisältöjä
 - Ymmärtäminen **esimerkkien** kautta (fantasia ja ohjelmistotestaus)
 - **Tutkiva** testaus (kriittinen ajattelu)

Lasten inspiroima tutkivan testauksen tarkistuslista

Helppo

- Hauskaa?
- Outoa?
- Minäkin voin tehdä tämän!
- Hidastaa minua?
- Voisi olla helpompi?
- Väärinymmärrys?
- Ruma?
- Huijausta?
- Kuten odotettu?

Edistynyt

- Käy läpi kaikki polut
- Peruuta
- Tarkista ohjeita vasten
- Voimmeko molemmat tehdä samoin?
- Koordinoi rinnakkainen testi
- Erikoismerkit
- Profiilit ja data

Kirjan rakenne

- Kirjan voi lukea sekä fantasiatarinana että ohjelmistotestauksen oppina harjoituksineen
 - Tekstilaatikot tarinan seassa kertovat koodauksen ja testauksen ajatuksia
- Luvut on nimetty vikatyypin mukaan (eli lohikäärmeet)
 - Kyseisen vikatyypin huomaamiseen sopiva testaustyyppi esitellään myös
 - Ohjelmistokehityksen ja –testauksen ideoita laajennetaan luku luvulta
- Jokaisen luvun lopussa on harjoituksia
 - Myös oikeita / mallivastauksia
- Joka luvun lopussa on lohikäärmeen hahmolomake
 - Pelillistämistä
 - Yhteen veto
- Sanasto kirja lopussa



89

Harjoitukset ja vastaukset

EXERCISES

Exercise 11.1 Which defect?

Which of these problems may be a functional defect?

1. There are no longer any t-shirts for sale in the online store.
2. The value of the shopping cart increases with less than the price of the selected purchase.
3. Payment options don't include a credit card.
4. The online store seems slow.

Exercise 11.2 How do you test this?

You're a tester. The value of the shopping cart is less than the sum of the prices of the products you select. How do you investigate the situation?

Exercise 11.3 How does the story continue?

You're Tom. You have just heard from Laura that many small dragons have gone through the hole in the wall to the village. You go to warn the villager waiting in the pit, but you stumble at the edge of the pit. You fall on top of the villager and the twigs. You create quite a noise, and additionally, the villager cries out loud because your head directly hits his teeth. The dragons hear the noise and run through the hole in the wall to the forest.

What could happen next in the story? What would that mean in the world of software testing?

Exercise 11.4 Let's play!

Take out some grid paper and pencils. Ask a friend to help. Let's play the dragon version of the battleship game. Each player draws two 10x10 grids on their paper, and number each grid with 1-10 for rows and A-I for columns. Then, secretly from the other, each player draws a dragon in the grid by darkening all the edges of the used squares. The dragon is five squares long, but you can bend it. It is enough for the dragon squares to touch each other at one corner. The second grid is left blank, as it is the opponent grid.

Let's play. The younger one starts. You try to guess the position of the other player's dragon squares by saying, for example, C4, meaning C-column and 4-row. Just one guess. The other player answers whether or not it hits. You should mark your successful guess with a cross on your opponent grid and a missed guess with a circle. Another player marks the hits and misses in their dragon grid, respectively. Then it's the second player's turn to guess once where the first player's dragon is. The players continue until one of the dragons is killed, that is, all five squares are hit. Remember that there are often more defects near one another. The dragon is in adjacent squares!

ANSWERS TO EXERCISES

Exercise 11.1 Which defect?

1. Not a functional defect. The online store has no more t-shirts for sale.
2. Yes, it is a functional defect. The calculation of the shopping cart value is wrong.
3. Not a functional defect. There is no credit card option, that's all. If it used to be there, then this is a version control defect.
4. Not a functional defect, but a performance defect.

Exercise 11.2 How do you test this? – Sample answer

Based on the symptoms, the problem appears to be a functional defect in the calculation. You test the calculation one product at a time. You select shoes to the cart. Is the shopping cart value correct? It should be the price of the shoes. You also add socks to the cart. Is the shopping cart value correct? Reduce the cost of shoes in the test data, and do the same for the price of socks. Repeat the tests. Does the shopping cart now calculate prices correctly? Try whole dollars for shoes and socks. Then try using dollars and cents. Perhaps the value of the cart only shows full dollars. So, there is a rounding error in the calculation!

Exercise 11.3 How does the story continue? – Sample answer

You are very annoyed. If you just hadn't stumbled. You shoot at one dragon and hit it. Other dragons flee to the forest. You drag the dead dragon into the village to investigate. You learn where to hit a dragon. Although you didn't catch all the dragons, they may still be back. It is worth trying the plan again. Now only more cautiously, without making a sound. At least you can shoot the dragons in the right spot.

In the world of testing, you should not stop testing when you have found one defect. Many times a faulty part of the software has additional defects. They may be similar to the defect you already found. So, continue testing the defective part until you no longer find any defects. Then you probably have found most of the defects.

1. Monivalintakysymys perusteluineen
2. Ohjelmistotestauskysymys liittyen annettuun skenaarioon
3. Luova tarinan jatkamista
4. Piirtämis – ja peliharjoituksia pelillistämismielessä ja hauskuuden nimissä

Hahmolomakkeiden esimerkkejä

Dragon Grey robber



- Colour: Grey
- Size: Large
- How hard to find?: Easy
- How hard to vanquish?: Medium
- Flies?: No
- Wings: Small
- Breathes fire?: Yes
- Favourite things: Stealing food and treasure

Defect

Functional defect

- **Defect type:** Functional defect
- **Severity:** Medium
- **Visibility:** High
- **Symptoms of the defect:** The software doesn't do what it should. The user sees the information in the wrong place.
- **Cause of the defect:** Functionality has been coded wrong.
- **Root cause:** The developer has misunderstood what the user wanted. Or the defect is because of carelessness, done in a hurry.
- **Testing:** You use the software normally, based on tester experience or requirement definition.
- **Fixing:** You change the code to work correctly.

Dragon Clever green



- Colour: Glittering green
- Size: Small
- How hard to find?: Medium
- How hard to vanquish?: Medium
- Flies?: No
- Wings: Small
- Breathes fire?: Yes
- Favourite things: Stealing food and treasure.

Defect

Small but tricky defect

- **Defect type:** Functional defect
- **Severity:** Medium
- **Visibility:** Low
- **Symptoms of the defect:** The software doesn't do what it should. The problem is tricky, although not that big. The user sees the information in the wrong place.
- **Cause of the defect:** Functionality has been coded wrong.
- **Root cause:** The developer has misunderstood what the user wanted. Or the defect is because of carelessness, done in a hurry.
- **Testing:** You use the software normally, based on tester experience or requirement definition.
- **Fixing:** You change the code to work correctly.

Dragon Two-headed grey



- Colour: Grey
- Size: Large
- How hard to find?: Easy
- How hard to vanquish?: Medium
- Flies?: Yes
- Wings: Large
- Breathes fire?: Yes
- Favourite things: Stealing food and treasure.

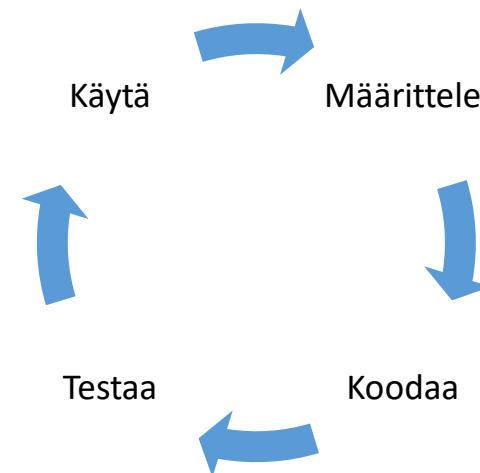
Defect

A really bad defect

- **Defect type:** Functional defect
- **Severity:** High
- **Visibility:** High
- **Symptoms of the defect:** The software doesn't do what it should. The problem is big and severe. The user sees the information in the wrong place.
- **Cause of the defect:** Functionality has been coded wrong.
- **Root cause:** The developer has misunderstood what the user wanted. Or the defect is because of carelessness, done in a hurry.
- **Testing:** You use the software normally, based on tester experience or requirement definition.
- **Fixing:** You change the code to work correctly.

Opetuksessa katettuja asioita

- Ohjelmistotestauksen & ohjelmistokehityksen perusasioita
- Ohjelmistokehityksen elinkaaria: uuden ohjelmiston kehitys, vanhan ohjelmiston ylläpito, ketteryys, DevOps
- Erilaisia liiketoiminta-alueita, joilla ohjelmistoja käytetään
- Erilaisia tilanteita milloin ja miten testataan
- Ohjelmistojen vikatyyppejä
- Testaustyypppejä
- Testiautomaatio
- Regressiotestaus



Esimerkki kirjan käytöstä

- Oppitunnit 1-2 maksuttoman Creative Commons esityksen kautta, esittely aiheeseen
- Kirjan lukua kappale kerrallaan
 - Oppitunnilla, ennen, tai jälkeen
- Oppitunnit 3-15 (tunti per kappale)
 - (luetaan luku yhdessä)
 - Keskustelua luvun vikatyypistä ja testaustyypistä
 - Luvun harjoitusten tekeminen yhdessä
 - Testataan oppilaiden kännyköillä luvun mukaista testausta



Kiitos!

Dragons Out tietoisku

<https://www.dragonsout.com/p/dragons-out-tietoisku.html>

Tilaa kirjoja:

<https://www.dragonsout.com/p/order-dragons-out-book.html>

Seuraa kirjaprojektia:

- <https://www.dragonsout.com>
- <https://www.facebook.com/DragonsOutOy>
- <https://www.instagram.com/dragonsoutbook/>
- <https://twitter.com/DragonsOutOy>
- <https://www.linkedin.com/company/dragons-out/>

Kysymyksiä:

kari.kakkonen@dragonsout.com

