



DRAGONS OUT

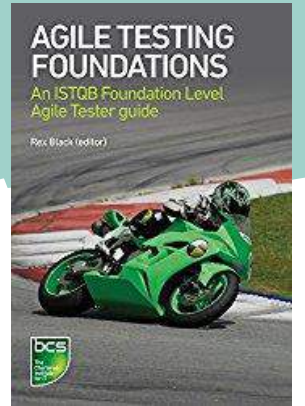
The book about dragons, knights and software testing

KARI KAKKONEN



Twitter: @kkakkonen

LinkedIn: fi.linkedin.com/in/karikakkonen/



• ROLES

- Knowit Oy, Director/Quality and Competences, Lead Consultant, Trainer and Coach
- Secretary of ISTQB Executive Committee
- Treasurer of Finnish Software Testing Board (FiSTB)
- **Children's and testing author at Dragons Out Oy**

• ACHIEVEMENTS

- 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 Full & Advanced Full & Agile Tester certified
- DASA DevOps Fundamentals, Scrum Master and SAFe certified
- SPICE provisional assessor certified
- M.Sc.(Eng), Helsinki University of Technology (present Aalto University), Otaniemi, Espoo

5.9.2019

© Dragons Out Oy

- 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 + Knowit Quality Professional
- DASA DevOps
- Quality & Test process and organization development, Metrics
- Agile testing, Scrum, Kanban, Lean
- Leadership
- Test automation, Mobile, Cloud, DevOps
- Quality, Cost, Benefits.



KNOWiT HAS OFFICES

in Sweden, Norway, Finland,
Denmark and Germany



THE COMPANY IS LISTED

on the Nordic Exchange in Stockholm



ESTABLISHED

in 1990



301 000 000 €

in net sales 2018

CREATING
DIGITAL
OPPORTUNITIES

knowit

MORE THAN 2 200

specialist in total.
In Helsinki office
approx. 130 specialists.



KNOWiT SERVICES

- Quality Software Development
- Intelligent Automation
- Software Development Consultancy
- Training and Coaching
- Client Experience offering

WWW.KNOWiT.FI





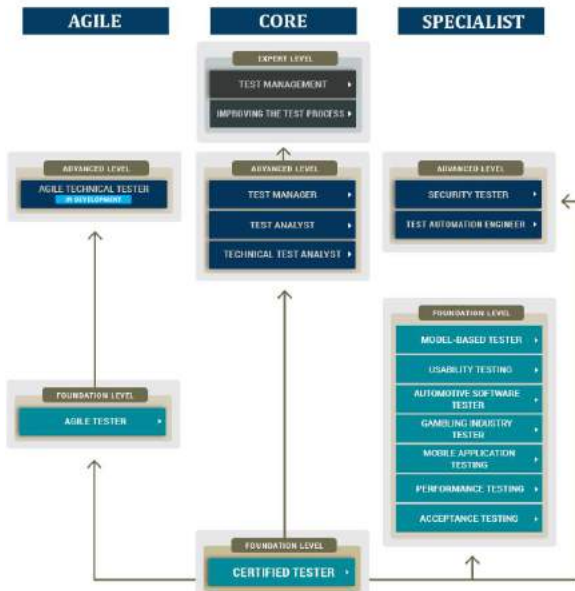
Number of exams administered: **~875,000**

Number of certifications issued: **641,000+**

In Over 120 countries



 Countries covered by **Member Boards and Global Exam Providers**  Countries covered by **Global Exam Providers**





Testaus nuorille -

miten innostaa testausuralle jo varhain?



Agenda

- Koodauksen lisäksi myös testaukseen pitää innostaa lapsia / nuoria
- Lähestymistapa pitää olla hauska ja tarinanomainen
- Lohikäärme- ja ritaritarinoiden tarjoamat analogiat testaukseen
- Suunnittele oma lohikäärmeesi (eli bugisi)



Miksi testausta nuorille / lapsille?

- Koodaus on alkanut kiinnostaa jo lapsia ja nuoria.
- Monet tahot yhteiskunnassa panostavat koodikouluihin yms.
- Ohjelmistojen testausta ei opeteta riittävästi.
- Testaus on jopa yli puolet kaikesta ohjelmistokehitystyöstä.
- **Myös testaajista on pula, ei vain koodaajista.**
- Testaus on uusi perustaito
- Tarvitaan hyvää laatua ohjelmistojen kehitykseen Suomessa ja maailmalla
- Tarvitsee siis tehdä testaus tutuksi jo lapsille
 - Testauskouluja
 - Testauskirjoja
- Oma ratkaisuni on kaikkien koodikoulujen lisäksi lapsille suunnattu kirja ohjelmistotestauksesta.



Miten lapset ja nuoret oppivat?

- Tarinan kautta
- Esimerkin kautta
- Yksinkertaistus
- Selkeys
- Kokeilemalla, tekemällä
- Toistamalla
- Muistamalla
- Samaistumalla
- Yrittämällä ja erehtymällä





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

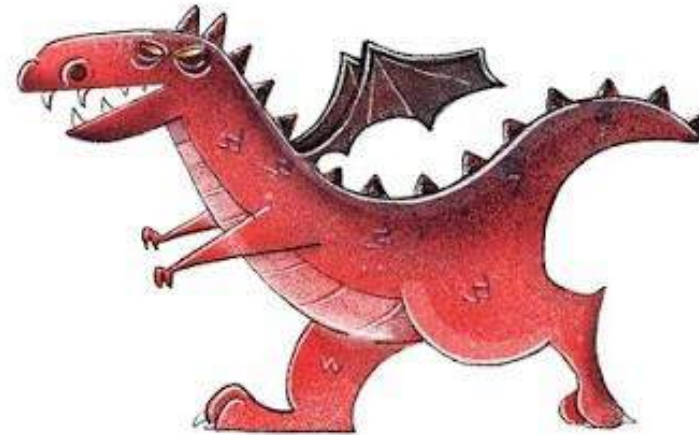
- *Tarinassa lohikäärme saapui kylään kesken paaluvarustuksen kunnostamisen. Vastaavasti suurin osa vioista ohjelmistoissa löytyy kehityksen aikana ennen kuin ohjelmisto on valmis. Silloin **vikojen löytäjät (testaajat) ja korjaajat (koodaajat)**, ovat koko ajan saatavilla. Yleensä testaaja löytää viat, 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

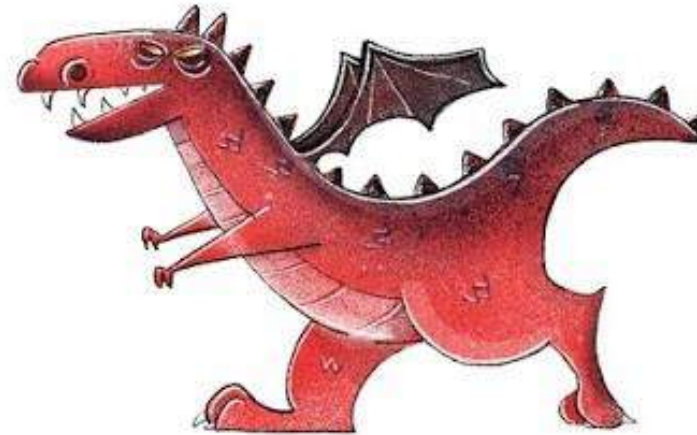
- **Väri:** Punainen
- **Koko:** Keskitaso
- **Vaikeus löytää:** Vaikea
- **Vaikeus karkottaa:** Helppo
- **Lentää?:** Ei
- **Siivet:** Pienet
- **Syöksee tulta?:** Kyllä
- **Lempipuuha:** Lampaiden syönti





Ärsyttävä lohikäärme

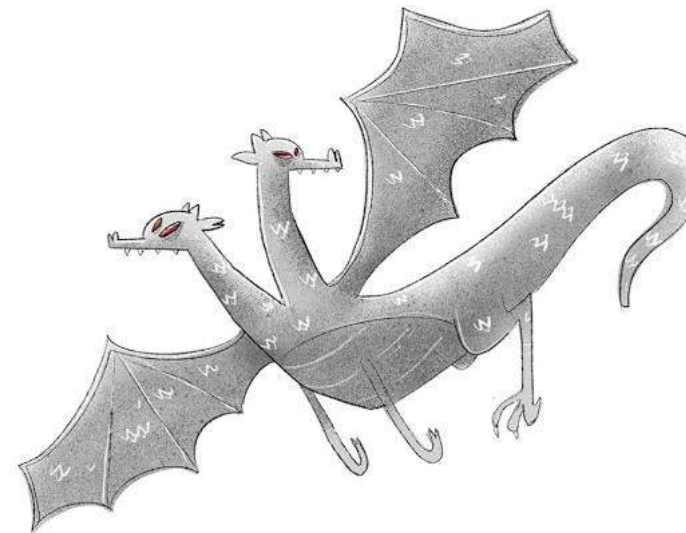
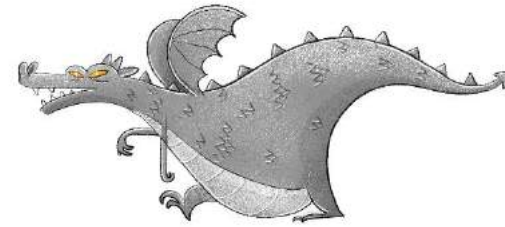
- **Vian nimi:** Muistivuoto
- **Vakavuus:** Keskitaso
- **Vian oire:** Laite hidastuu, kunnes se ei voi enää tehdä mitään ja se sammuu
- **Vian syy:** Muistia varataan ohjelmiston käyttöön, mutta käytön jälkeen sitä ei vapauteta
- **Alkuperäissyitä:** Koodaaja ei ole huolellinen muistinvapauttamisessa. 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.





Ryöstelevät lohikäärmeet

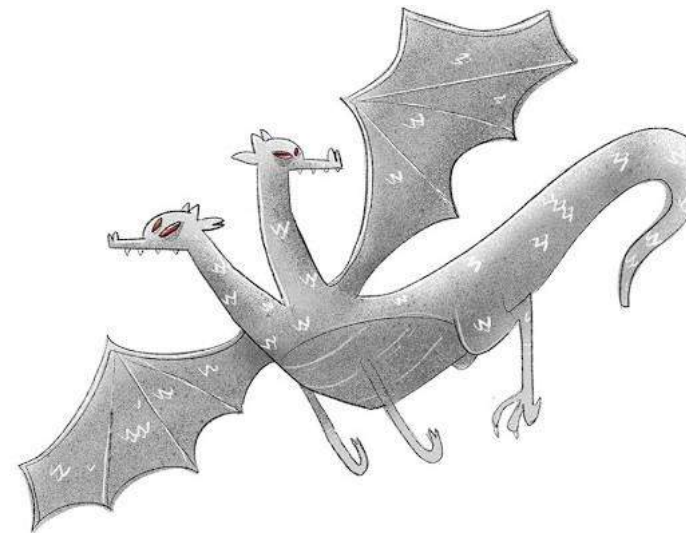
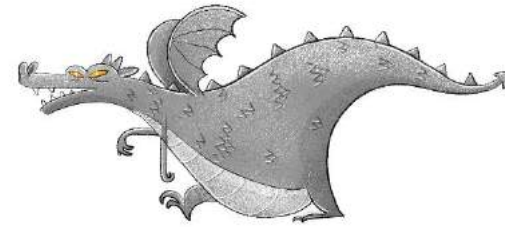
- **Väri:** Vihreän kimaltava tai harmaa
- **Koko:** Pienestä isoon
- **Vaikeus löytää:** Helposta vaikeaan
- **Vaikeus karkottaa:** Helposta vaikeaan
- **Lentää?:** Osaa lentää, osa ei
- **Siivet:** Pienistä isoihin
- **Syöksee tulta?:** Kyllä
- **Lempipuuha:** Ruoan ja aarteiden ryöstäminen





Ryöstelevät lohikäärmeet

- **Vian nimi:** Toiminnallisuusviat
- **Vakavuus:** Matala-Keskitaso-Suuri
- **Vian oire:** Ohjelmisto ei tee sitä, mitä pitäisi. Laskutoimituksesta tulee väärä tulos. Käyttäjä näkee väärän tiedon väärässä paikassa.
- **Vian syy:** Toiminnallisuus on koodattu väärin.
- **Alkuperäissyitä:** Koodaaja on ymmärtänyt väärin , mitä käyttäjä on pyytänyt. Tai vika on syntynyt huolimattomuudesta, kiireessä.
- **Testaus:** Ohjelmistoa käytetään normaalisti, perustuen testaajan kokemukseen tai vaatimusmäärittelyyn.
- **Korjaus:** Koodia muutetaan toimimaan oikein.





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:** Tietoturvakatastrofi
- **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 tietoturvakatastrofia ja murtautunut järjestelmään, sitten varastanut tai tuhonnut jotain.
- **Alkuperäisyyttä:** Koodaaja ei ole noudattanut uusimpia tietoturvalinjojen koodauksen ohjeita. Ehkä ei osaa tai tiedä.
- **Testaus:** Etsitään ohjelmistosta tunnettuja tietoturvakatastrofia käyttämällä ohjelmistoa itse tai tietoturvaohjelman avulla. Voi myös tutkia koodia. Apuna lista tunnettuja vikoja.
- **Korjaus:** Tunnettuun vikaan liittyy myös tunnettu korjaus. Se tehdään koodin, tai järjestelmän asetuksiin.





Nopea lohikäärme

- **Väri:** Purppura
- **Koko:** Pieni
- **Vaikeus löytää:** Vaikea
- **Vaikeus karkottaa:** Vaikea
- **Lentää?:** Kyllä
- **Siivet:** Suuret
- **Syöksee tulta?:** Vähän
- **Lempipuuha:** Kullan ryöstäminen salassa





Nopea lohikäärme

- **Vian nimi:** Suorituskykyvika
- **Vakavuus:** Keskitaso
- **Vian oire:** Ohjelmisto toimii hitaammin kuin pitäisi.
- **Vian syy:** Koodin osa toimii huonosti tai väärin, tai järjestelmän asetukset ovat väärässä. Seurauksena ohjelmisto yrittää tehdä jotain turhaa tai väärää. Aikaa kuluu.
- **Alkuperäisyyttä:** Nopeutta ei ole ajateltu koodatessa. Ei osata käyttää ohjelmointiympäristön parhaita ominaisuuksia. Ehkä ei nähdä oman koodin ulkopuolelle muuhun järjestelmään.
- **Testaus:** Käytetään järjestelmää normaalisti, yksi tai useampi käyttäjä kerrallaan, usein suorituskykyohjelmiston avulla. Mitataan nopeutta eli vasteaikaa.
- **Korjaus:** Muutetaan niitä koodin kohtia, jotka toimivat hitaasti. Kokeillaan korjausta ja mitataan nopeutta uudestaan.





Näyttävä, hankala lohikäärme

- **Väri:** Vihreä
- **Koko:** Suuri
- **Vaikeus löytää:** Helppo
- **Vaikeus karkottaa:** Vaikea
- **Lentää?:** Kyllä
- **Siivet:** Suuret
- **Syöksee tulta?:** Paljon
- **Lempipuuha:** Kullan vaaniminen ja oman reviirin puolustus





Näyttävä, hankala lohikäärme

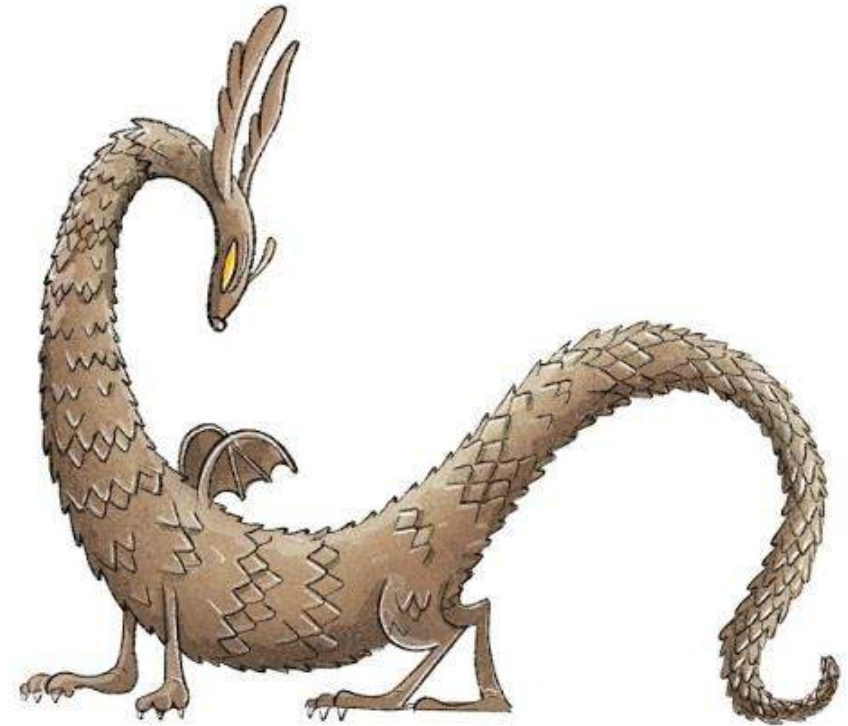
- **Vian nimi:** Käytettävyyssvika
- **Vakavuus:** Matala
- **Vian oire:** Ohjelmistoa on hankala käyttää, mutta se onnistuu.
- **Vian syy:** Koodaus on tehty vain toiminnallisuutta ajatellen, mahdollisesti helpoimmalla tavalla.
- **Alkuperäisyyttä:** Käyttäjien tarpeita ei ole mietitty suunnitellessa tai koodatessa. Käytettävyyttä ei ehkä ymmärretä.
- **Testaus:** Käytetään järjestelmää normaalisti. Kerätään mielipiteitä, mitä on helppo käyttää ja mitä vaikea käyttää.
- **Korjaus:** Muutetaan niitä koodia helpommin käytettävään suuntaan käyttäjätestauksen toiveiden mukaisesti.





Maanalainen lohikäärme

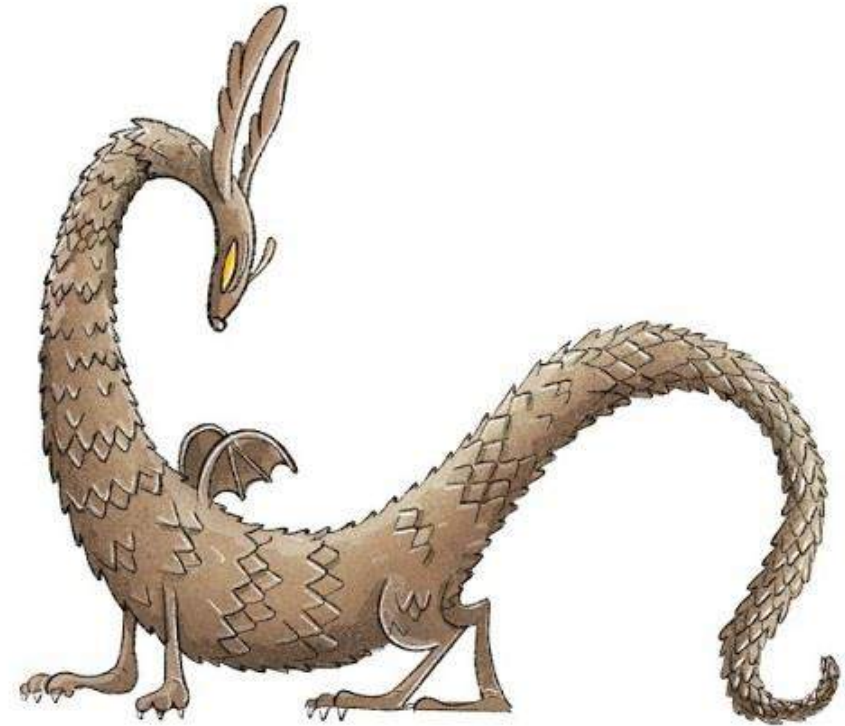
- **Väri:** Ruskea
- **Koko:** Suuri
- **Vaikeus löytää:** Helppo
- **Vaikeus karkottaa:** Keskitaso
- **Lentää?:** Ei
- **Siivet:** Pienet
- **Syöksee tulta?:** Paljon
- **Lempipuuha:** Helpon ruoan etsiminen ja syöminen





Maanalainen lohikäärme

- **Vian nimi:** Laitteistovika
- **Vakavuus:** Suuri
- **Vian oire:** Jokin osa tietokoneesta ei toimi tai tietokone ei toimi ollenkaan.
- **Vian syy:** Laitteiston osa on hajonnut ajan kuluessa.
- **Alkuperäisyyttä:** Laitteiston osa saattoi olla huonolaatuinen, eikä kestänyt niin pitkään kuin olisi pitänyt. Ehkä osa ei sovi muihin osiin kunnolla, jonka vuoksi se hajoaa.
- **Testaus:** Käytetään järjestelmää normaalisti. Tarkkaillaan laitteen eri osia. Testausympäristössä käytetään samanlaisia osia kuin mitä käyttäjille tulee olemaan.
- **Korjaus:** Vaihdetaan osa ehjäksi tai vaihdetaan paremmin sopiva osa viallisen tilalle.





Varjo vai lohikäärme?

- **Väri:** Harmaa
- **Koko:** Pieni
- **Vaikeus löytää:** Helppo
- **Vaikeus karkottaa:** Helppo
- **Lentää?:** Ei
- **Siivet:** Pienet
- **Syöksee tulta?:** Kyllä
- **Lempipuuha:** Ihmisten hämääminen





Varjo vai lohikäärme?

- **Vian nimi:** Testattavuusvika
- **Vakavuus:** Pieni
- **Vian oire:** Toiminnallisuus näyttää toimivan väärin, esim. väärä laskun lopputulos.
- **Vian syy:** Testaajalla on ollut vaikka väärä testiaineisto tai väärä testiympäristö.
- **Alkuperäisyyttä:** Testaaja voi olla liian innokas löytämään vikoja. Testausympäristöä ja testiaineistoa ei ole suunniteltu riittävän tarkasti.
- **Testaus:** Testataan normaalisti, mutta kiinnitetään huomiota ympäristöön ja aineistoon. Harkitaan aina onko testaus itse vian syy.
- **Korjaus:** Määritellään testausympäristö ja –aineisto paremmin. Opitaan vääristä hälytyksistä.





Kiltti lohikäärme

- **Väri:** Vihreän kimaltava
- **Koko:** Keskitaso
- **Vaikeus löytää:** Helppo
- **Vaikeus karkottaa:** Helppo
- **Lentää?:** Kyllä
- **Siivet:** Keskitaso
- **Syöksee tulta?:** Kyllä
- **Lempipuuha:** Eläinten syöminen ja ihmisten auttaminen





Kiltti lohikäärme

- **Vian nimi:** Vian kylväminen – tahallaan tehty vika
- **Vakavuus:** Pieni
- **Vian oire:** Toiminnallisuus näyttää toimivan väärin, esim. väärä laskun lopputulos. Vika vaikuttaa siis toiminnallisuusvialta.
- **Vian syy:** Testaaja tai koodaaja on tehnyt vian koodiin tarkoituksella.
- **Alkuperäisyyttä:** Ajatus on, että kun kylvetyt viat on löydetty, on löydetty kaikki viat.
- **Testaus:** Testataan normaalisti, ja yritetään löytää kaikki kylvetyt viat. Samalla löytyy oikeitakin vikoja. Kun viimeinen kylvetty vika on löydetty, testaus voidaan lopettaa.
- **Korjaus:** Muistetaan korjata koodista myös kaikki kylvetyt viat, kuten oikeatkin toiminnallisuusviat





Sitkeä lohikäärme

- **Väri:** Vihreän kimaltava
- **Koko:** Suuri
- **Vaikeus löytää:** Helppo
- **Vaikeus karkottaa:** Vaikea
- **Lentää?:** Kyllä
- **Siivet:** Pienet
- **Syöksee tulta?:** Kyllä
- **Lempipuuha:** Ihmisten häiriköinti toistuvasti





Sitkeä lohikäärme

- **Vian nimi:** Versionhallintavika
- **Vakavuus:** Keskitaso
- **Vian oire:** Aiemmin korjattu vika löytyy uudelleen, toisin sanoen vaikkapa toiminnallisuusvika.
- **Vian syy:** Versionhallinta on pettänyt. Koodaaja ottaa käyttöön sellaisen vanhan osan ohjelmistoa, josta on olemassa uudempi osa. Tuohon uuteen osaan joku toinen koodaaja on jo tehnyt vikakorjauksen.
- **Alkuperäisyyttä:** Kiire ja liian suuri henkilömäärä tuottavat vikoja. Versionhallinta voi puuttua.
- **Testaus:** Testataan normaalisti. Jos vika löytyy, tarkistetaan, onko sellainen jo korjattu. Jos on, tutkitaan versionhallinnan toimintaa.
- **Korjaus:** Parannetaan versionhallinnan toimintatapaa. Otetaan käyttöön versionhallintaohjelmisto.





Erehtyvä lohikäärme

- **Väri:** Vihreän kimaltava
- **Koko:** Pieni
- **Vaikeus löytää:** Helppo
- **Vaikeus karkottaa:** Helppo
- **Lentää?:** Kyllä
- **Siivet:** Keskitaso
- **Syöksee tulta?:** Kyllä
- **Lempipuuha:** Isojen lohikäärmeiden hännäminen





Erehtyvä lohikäärme

- **Vian nimi:** Dokumentaatiovika
- **Vakavuus:** Matala
- **Vian oire:** Toiminnallisuus toimii eri tavalla kuin ohjeiden mukaan pitäisi toimia.
- **Vian syy:** Suunnitelmat ja ohjeet eivät vasta koodattua ohjelmistoa
- **Alkuperäisyyttä:** Koodauksen aikana on tehty päätöksiä tai tulkintoja, jotka muuttavat ohjelmistoa erilaisiksi kuin on suunniteltu. Ohjeet on tehty vanhojen suunnitelmien pohjalta.
- **Testaus:** Testataan normaalisti. Jos vika löytyy, pohditaan onko ohjelmisto vai ohje väärässä.
- **Korjaus:** Muutetaan joko ohjelmistoa tai ohjetta, jotta ne vastaavat toisiaan.





Ritarit

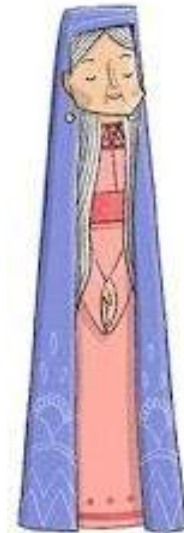
- Kehittäjät
 - Koodaaja
 - Testaaja
- Toimivat yhdessä, yleensä samassa kehitystiimissä (Ketteryys)
- Rakentavat ohjelmistoja
- Testaavat ja korjaavat vikoja





Lapset, kyläläiset

- Käyttäjiä
- Auttavat rakentamaan ohjelmistoja
- Testaavat uusia ohjelmistoja
- Testaavat vanhoja ohjelmistoja
- Pyytävät apua tarvittaessa ylläpitäjiltä tai kehittäjiltä





Metsästäjät

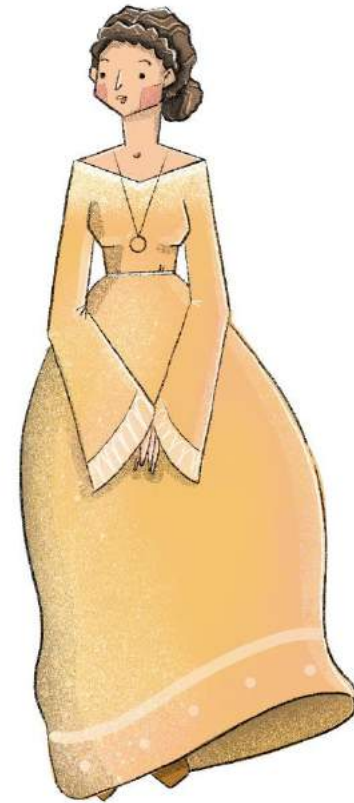
- Ylläpitävät järjestelmää
- Testaavat
- Poistavat vikoja
- Auttavat käyttäjiä
- Pyytävät tarvittaessa apua kehittäjiltä
- Joskus mukana kehitystiimissä (DevOps-tiimi)





Linnanherrat ja linnanrouvat

- Tilaavat ohjelmistoja ja järjestelmiä
- Tuote-omistajia
- Johtoa
- Määrittelevät, mitä järjestelmissä pitää olla
- Kuuntelevat kehittäjiä





Tietäjät

- Asiantuntijoita
 - Käytettävyydessä
 - Tietoturvassa
 - Suorituskyvyssä
- Auttavat tilaajia
- Auttavat kehitystiimiä





Oman lohikäärmeen suunnittelu

- Tarvikkeet
 - Paperia ja kyniä
- Tehtävä
 - Muistele eri lohikäärmeitä / vikoja
 - Keksi lohikäärme, jonka haluat piirtää
 - Ehkä jokin vika, johon olet itse törmännyt
 - Pelkistä
 - Mitkä ovat lohikäärmeen / vian ominaisuudet? Iso, pieni? Vaikea löytää?
 - Piirrä
 - Lohikäärme paperin keskelle
 - Kirjoita toiseen laitaan lohikäärmeen ominaisuudet
 - Kirjoita toiseen laitaan vian ominaisuudet





Mitä seuraavaksi?

- Ajatuksesi saattavat päätyä mukaan Dragons Out kirjaan
 - Kirja lohikäärmeistä, ritareista ja ohjelmistotestauksesta
 - Lapsille, nuorille, aikuisille
 - Hauska tapa oppia ohjelmistotestauksesta
- Tehdään yhdessä ohjelmistotestaus tutuksi uusille sukupolville!
- Kirja on nyt joukkorahoituksessa Indiegogo:ssa
<https://igg.me/at/dragons-out-book>
 - Ennakko-osta kirjoja tai muita etuja
 - Kerro kirjasta kaikille!





Kirjasta ”Dragons Out”

- Kirjoittaja Kari Kakkonen
- Kuvittaja Adrienn Széll
- Tekstit ja kuvitus oikeudet Dragons Out Oy
- Tästä esityksestä tulee myöhemmin Creative Commons -lisensioitu versio jakeluun kirjan verkkosivustolle
- Lisätietoja www.dragonsout.com





DRAGONS OUT

The book about dragons, knights and software testing