Hyvät ohjelmistotuotannon käytänteet

**Ongelma**: Toteutettu ohjelmisto ei olekaan asiakkaan toiveiden mukainen.  
**Syy**: Et kehittänyt iteratiivisesti, et saanut asiakkaan mielipiteitä kehityksen välissä.  
**Ratkaisu**: kehitä iteratiivisesti, kerää palautetta ja kysy asiakkaalta mielipiteitä.

**Ongelma**: Jokainen ohjelmoija toteuttaa osan ohjelmistosta, mutta lopuksi ilmenee, että palaset eivät sovi yhteen.  
**Syy**: Kommunikointivika sekä liian vähäinen tai huonosti toteutettu testaus. Kommunikoinnin epämääräisyys tuottaa vakavia väärinymmärryksiä ja vähäinen tai huono testaus peittää palasten yhteensopivuuden ongelmat.  
**Ratkaisu**: Kehitä iteratiivisesti ja testaten ohjelmaa

**Ongelma**: Vakavat ongelmat huomataan vasta liian myöhään.  
**Syy**: Ohjelman testausta ei tehty tarpeeksi tai tehtiin huonosti.  
**Ratkaisu**: Kehitä iteratiivisesti testaten mahdollisia ongelmia

**Ongelma**: Ohjelmistoa on vaikea ylläpitää tai laajentaa.  
**Syy**: Koodi on huonosti toteutettu. Ohjelmisto toteuttaessa ei ole käytetty laadunvarmistusta. Visuaalista mallintamista ei ole käytetty.  
**Ratkaisu**: Kirjoita laatukoodia (helposti luettavaa, ymmärrettävää ja ylläpidettävää). Laadunvarmistus. Myös visuaalinen mallintaminen auttaa koko tuotannon selkeyttämiseksi.

**Ongelma**: Ohjelmiston laatu (esim. suorituskyky, tietoturva) on huono.  
**Syy**: Tuotanto on toteutettu kiireellisesti tai huolimattomasti, työn laatu alhaista.  
**Ratkaisu**: Varmista ohjelmiston tuotannossa laatu

**Ongelma**: Tuotantoprosessissa ei saa tapahtua muutoksia – tai koko pakka hajoaa käsiin.  
**Syy**: Muutoksia ei voi estää, eikä välttämättä kannata, joten muutoksia ei ole hallittu hyvin.  
**Ratkaisu**: Muutokset huomioidaan, priorisoidaan ja sitten toteutetaan kun on paras aika.

**Ongelma**: Kehittäjät sotkevat epähuomiossa toistensa tekemisiä; ei pysytä perässä kuka muutti mitäkin, milloin tai miten.  
**Syy**: Kehittäjät eivät kommunikoi riittävästi. Projektin arkkitehtuuri ja tiimien, sekä kehittäjien tehtävät ei ole jaettu kunnolla.  
**Ratkaisu**: Jaa arkkitehtuuri alijärjestelmiin ja anna vastuu tietystä alijärjestelmästä yhdelle tiimille – älä myöskään anna tiimien vaikuttaa toistensa alijärjestelmiin. Komponenttiarkittehtuurin käyttö