**Ieteikumi paskaidrojošam tekstam (J.Borzovs, 2019)**

1. Programmatūras projekta organizācija

Ar šo jāsaprot attiecības starp projektā iesaistītiem darbiniekiem gan projekta grupas (project team) ietvaros, gan ārpus tās.

Atsvaidziniet savas zināšanas, pārlasot [ 1 ] CHAPTER 24: Project Management Concepts: People: The Software Team, 651-454.

Bagātīgu projekta organizācijas grafisku attēlojumu klāstu var redzēt <http://templatelab.com/organizational-chart-templates/> .

1. Kvalitātes nodrošināšana

Atsvaidziniet savas zināšanas, pārlasot [ 1 ] CHAPTER 16: Software Quality Assurance, 432-448 .

1. Konfigurāciju pārvaldība

Atsvaidziniet savas zināšanas, pārlasot [ 1 ] CHAPTER 22: Software Configuration Management, 584-612.

1. Darbietilpības novērtējums

Pareizāk būtu teikt – izstrādātā programmkoda “normatīvās” darbietilpības pamatojums.

Reference

1. Roger S. Pressman. Software Engineering: A Pratictioner’s Approach, 7th ed., 2010.

**Kā pamatot, ka programmkods ir vismaz 3 personmēnešu vērts? (J.Borzovs, 2022)**

Pasaulē atzītākā komercsabiedrība, kas nodarbojas ar programmatūras darbietilpības prognozēšanu (effort estimation), ir QSM (Quantitative Software Management) - <http://www.qsm.com/> .

Jau izstrādāta programmkoda normatīvās darbietilpības noteikšana (nejaukt ar konkrētajā gadījumā faktiski patērēto) ir darbietilpības prognozēšanai inversa darbība. Programmkoda rindiņu skaits un/vai faktiskais funkcijpunktu daudzums ir zināms, bet jānoskaidro, cik profesionāļu cik ilgā laikā šo programmkodu būtu varējuši izstrādāt. Saprotams, ka to var pateikt, vien balstoties uz liela skaita projektu datiem. QSM regulāri krāj informāciju par pabeigtiem projektiem un rēķina minimālās, maksimālās, vidējās vērtības un mediānas. To pašu dažkārt dara progresīvas programmizstrādes komercsabiedrības, tāpēc ir vērts noskaidrot, vai konkrētā gadījumā ir pieejami komercsabiedrības dati, kurus vēl papildus varētu izmantot aprēķiniem.

Viena no labām iespējām ir izmantot QSM atklātā piekļuvē izliktās etalontabulas - <http://www.qsm.com/resources/qsm-benchmark-tables> , veicot saprātīgas interpolācijas vai ekstrapolācijas.

Piemēram, tabulas “Business Systems Implementation Unit (New and Modified IU) Benchmarks” aile Quartile 1 rāda informāciju par 25% mazākajiem (t.i., pirmās kvartiles jeb ceturtdaļas) no 550 projektiem. Šādi projekti vidēji ilguši 3,2 mēnešus, vidēji prasījuši 965 personstundas, tos izstrādājuši vidēji 1,57 cilvēki, šādu projektu lieluma mediāna ir 1889 programmkoda loģiskās rindiņas personmēnesī, bet projektu lielums variējas starp 131 u 3115 loģiskām rindiņām (vienā programmkoda rindiņā mēdz būt vairākas instrukcijas, bet var arī viena instrukcija aizņemt vairākas rindiņas; tukšās un komentāru rindiņas netiek uzskaitītas).

Ja Jūsu izstrādātā programmkoda lielums ir tuvs trīskāršai mediānai – 3 x 1889 = 5667, recenzentam būs neiespējami pamatot, ka izstrādātā programmkoda normatīvā darbietilpība nesasniedz 3 personmēnešus. Ja nu esat pagaidām vēl mazpieredzējis programmizstrādātājs, “taisnīgāk” būtu par pamatu ņemt Quartile 1 *Range 50%* augšējo robežu, t.i., 3 x 466 = 1398 . Šī *Range 50%* augšējā robeža , ka šo apjomu sasniedz vien 25% no pirmās kvartiles projektiem.

Protams, ka īpaši sarežģīta programmkoda gadījumā lielums būs pamatoti mazāks.

Cita iespēja būtu attiecīgos skaitļus iegūt savā prakses organizācijā vai darba vietā.

Juris Borzovs

28.10.2022.