

Программ хангамж бүтээх өртөг тооцоход өгөгдөл олборлолтыг ашиглах нь

С.Уянга^{1*}, Н.Мөнхцэцэг¹, Ч.Наранмандал²

¹ Монгол Улсын Их Сургууль, Хэрэглээний шинжлэх ухаан, инженерчлэлийн сургууль, Мэдээлэл, компьютерийн ухааны тэнхим

² Жи-мобайл ХХК

*uyanga@seas.num.edu.mn

Хүлээн авсан: 2018.03.20, засварласан: 2018.09.17, зөвшөөрсөн: 2018.09.20

Хураангуй

Хэрэглэгчийн хэрэгцээ шаардлагыг бүрэн хангасан, өндөр чанартай программ хангамжийн бүтээгдэхүүнийг хямд зардлаар, богино хугацаанд амжилттай хөгжүүлэхэд эдийн засгийн тооцоо, төслийн удирдлага ихээхэн үүрэгтэй. Программ хангамжийн инженерчлэлд программ хангамжийн өртөг, үр ашгийн шинжилгээ, тооцоо хийх, үнэлэх, тэр дундаа түүнийг бүтээх өртгийг бодитой тооцоолж, таамаглах нь чухал ач холбогдолтой. Энэхүү өгүүлэлд программ хангамжийн өртөг тооцоолох аргачлал, программ хангамжийг бүтээх өртгийг таамаглахад өгөгдөл олборлолтыг ашигласан нэгэн шийдлийг авч үзсэн.

Түлхүүр үг: Программ хангамжийг бүтээх өртөг, СОСОМО, өгөгдөл олборлолт, функцийн цэг, эшалгуур үзүүлэлт, Argioi алгоритм, WEKA

1. Удиртгал

Өдгөө байгууллагын хөгжил, стратеги төлөвлөлт, өрсөлдөх чадварт мэдээлэл, харилцаа холбооны технологийг чухалчлан авч үзэх болжээ. Мэдээллийн багтаамж, технологийн хүчин чадал асар хурдацтай өсөж буй өнөө үед мэдээллийн технологийн хэрэглээний хамтаар их хэмжээний өгөгдөл, түүнийг боловсруулах мэдээллийн системийн цогц шийдлийг авч үзэх учиртай. Ялангуяа программ хангамжийн амжилттай хөгжүүлэлт, түүний чанарын асуудлыг эдийн засгийн үр ашиг, хөрөнгө оруулалтын үр өгөөжтэй шууд холбон үзэх болсон. Үүнд программ хангамжийн хөгжүүлэлтийн аргыг оновчтой сонгох зайлшгүй шаардлага тавигдаж байна. Программ хангамж үйлдвэрлэгчдийн үндсэн зорилго бол хэрэглэгчийн шаардлагыг бүрэн хангасан чанартай бүтээгдэхүүнийг хамгийн хямд зардлаар, төлөвлөсөн хугацаанд захиалагч, хэрэглэгчид хүргэх явдал юм. Ингэхдээ эдийн засгийн тооцоо, тэр дундаа программ хангамж бүтээх өртгийг үндэслэлтэй зөв тооцоолж, нөөцийг оновчтой удирдах нь чухал.

Программ хангамжийг бүтээх өртөг гэдэг нь тухайн программ хангамж, төслийн үйл ажиллагааны барагцаалсан өртгийн хэмжээ юм. Бүтээх өртгийн хэмжээ нь программын кодын мөр, функцийн цэгийн хэмжээнээс гол төлөв хамаардаг (Benala T.R., 2014).

Программ хангамж хөгжүүлэлтийг хэд хэдэн үйл ажиллагаанд хуваан үзэж, түүнийг бүтээх өртгийн тооцооллыг дор дурдсан асуултад хариулт авах зорилгоор хийдэг. Үүнд:

1. Үйл ажиллагаа бүрийг дуусгахад шаардлагатай хүчин чармайлт (хүн/сараар илэрхийлнэ.)
2. Үйл ажиллагаа бүрийг дуусгахад шаардлагатай хугацаа
3. Үйл ажиллагаа тус бүрийн өртөг.

Программ хангамж хөгжүүлэх нийт өртөгт дараах зардлыг авч үзнэ [1]. Үүнд:

1. Техникийн үйлчилгээг багтаасан техник хангамж болон программ хангамжийн зардал
2. Чадавхжуулалт, сургалтын зардал
3. Хөгжүүлэгчдийн хүчин чармайлтыг үнэлсэн зардал (Capt Gregory E.Brown, April 2017).

Программ хангамжийн үнэлгээг түүнийг хөгжүүлэх өртөг болон ашгийн нийлбэр хэмээн үзэж болно.



Зураг 1 . Программ хангамжийг бүтээх өртгийг тооцох үйл ажиллагаа

Бүтээх өртөг тооцох үйл ажиллагаа нь тодорхой алхамтайгаар хийгдэнэ. Программ хангамжийн хэмжээг тооцож, шаардагдах хичээл зүтгэлийг тооцоолсноор түүний зардал болон төслийн хугацааг тодорхойлох боломжтой болно. Өртөг тооцох үйл ажиллагаа ба программ хангамж хөгжүүлэх амьдралын циклийн хамаарах хамаарлыг Зураг 2 дээр харуулав.



Зураг 2. Программ хангамжийг бүтээх өртгийг тооцох нь

Программ хангамжийг бүтээх өртгийг түүний хөгжүүлэлтийн амьдралын циклийн эхэн үед урьдчилсан байдлаар тооцоолдог. Улмаар хөгжүүлэлтийн (төслийн) үе шат бүрд шинжлэн тооцоолох замаар тухайн төслийн бодит өртгийг тодорхойлдог (Dr.Ulbert, 2014) (Зураг 2).

2. Зардал тооцох аргууд

Зардлын тооцоо нь үйл ажиллагаа, арга гэсэн 2 хэсгээс бүрддэг. Энэ нь бодит болон нийт зардлыг тооцоход тусалдаг. Программ хангамж үйлдвэрлэгч нь хамгийн бага зардлаар (хямдаар), төлөвлөсөн хугацаанд чанартай бүтээгдэхүүнийг захиалагчид нийлүүлэх гол зорилготой ажилладаг. Программ хангамжийн зардлын тооцоо нь үндсэндээ системийн шинжээчид төсөлд шаардлагатай чухал нөөцүүд болон төслийн хугацааг нарийвчлалтайгаар авахад хэрэглэгддэг. Зардлыг тооцоолоход хэмжээ, цаг хугацаа, хичээл зүтгэл гэсэн үзүүлэлтүүдийг авч үзнэ. Зардлыг тооцох үйл ажиллагаа нь үндсэн 4 алхамтай. Хамгийн эхэнд программ хангамжийн хэмжээг тооцож, улмаар тухайн программ хангамжийг хөгжүүлэхэд шаардагдах хичээл зүтгэлийг тооцоолдог. Ингэснээр программ хангамжийг хөгжүүлэхэд шаардлагатай нийт зардал болон хөгжүүлэлтийн хугацаа гарна.

Программ хангамжийн зардлыг тооцоолоход төрөл бүрийн арга техник хэрэглэгддэг. Эдгээр аргыг гол төлөв алгоритмын болон алгоритмын бус гэж хоёр ангилдаг [4]. Алгоритмын арга нь математикийн тэгшитгэл дээр үндэслэх ба ийм аргуудаас COCOMA (Constructive Cost Model)

арга хамгийн түгээмэл хэрэглэгддэг. Алгоритмын аргууд нь олон давуу талтай боловч тэдгээрийг эзэмшиж, ашиглахад төвөгтэй, харьцангуй олон тооны өгөгдөл шаарддаг онцлогтой. Харин алгоритмын бус аргууд нь сурахад хялбар боловч тухайн программ хангамжийн төслийг түүнтэй ижил төстэй, өмнөх төслүүдтэй харьцуулж, тэдгээрийн өгөгдөл дээр тооцоолдог тул өмнөх төслүүдийн талаарх бүрэн мэдээллийг шаарддаг. Зардлыг тооцоолох дараах аргууд түгээмэл хэрэглэгддэг.

А. Алгоритмын бус аргууд

Алгоритмын бус аргууд нь адилтгах (analogy), хасах (deduction) үйл ажиллагаан дээр суурилж тооцоолдог. Иймээс хэмжээг нь тооцоолох программ хангамжаас өмнө хийгдэж байсан программ хангамжуудыг, тэдгээрийн ижил төстэй талуудыг мэдэх шаардлагатай байдаг. Зардлын үнэлгээ нь өмнөх программ хангамжийн төслийн үндсэн шинжилгээ эсвэл өгөгдлийн багц дээр хийгддэг. Алгоритмын бус онолын дараах арга, техник өргөн хэрэглэгддэг. Тухайлбал:

- *Адилтгалд суурилсан үнэлгээ.* Энэ нь зардлыг тооцоходоо тухайн төсөлтэй хамгийн ойр төстэй түүхт программ хангамжийг харьцуулж тооцдог. Өмнөх төслүүдийн өгөгдөл, үнийн дүнг харьцуулж тухайн төслийн зардлыг тооцоолон гаргадаг. Энэ аргыг мэдээллийн систем, эсвэл модулийн түвшинд аль алинд нь ашиглаж болно.

Адилтгал дээр үндэслэн тооцоход явагдах тодорхой үе шатуудтай. Үүнд:

1. Одоогийн төслийн шинж чанарыг тодорхойлох
2. Мэдээллийн санд хадгалагдаж буй шинж чанараараа ижил төстэй төслийг олох
3. Ижил төстэй төслүүдээс одоогийн төслийн зардлыг тооцоолох.

- *Мэргэжилтний үнэлэмжийн арга.* Энэ арга нь шинжээчийн мэдлэг, туршлагад түлхүү суурилдаг. Зардлын үнэлгээ нь тэдгээр төсөлд хэдэн шинжээч оролцоноос хамаарна. Мэдээллийг тодорхойлох, илрүүлэх, цуглуулах үйл ажиллагаа хязгаарлагдмал үед энэ аргыг ашиглах нь тохиромжтой. Программ хангамжийн төслийн стратегийг тооцоолоход өргөнөөр ашиглагддаг. Мэргэжилтний үнэлэмж арга дээр суурьлагдсан зардал тооцох арга бол Wide band Delphi арга юм.

- *Дээрээс-доош тооцооллын арга (Top-down Estimation).* Энэ арга нь алгоритмын болон алгоритмын бус аль аргад хэрэглэгддэг. Нийт зардлыг системийн хүрээнд гаргаад, улмаар тухайн зардлаа тухайн системийн бүрэлдэхүүн

хэсэг, модулиудад хуваагддаг. Энэ аргыг программ хангамжийн хөгжүүлэлтийн эрт үед илүү ашигтай. Энэ аргын жишээ бол Putnam загвар юм.

- *Доороос-дээш тооцооллын арга (Bottom-up Estimation)*. Энэ нь өмнө дурдсан дээрээс-доош тооцооллын үнэлгээний эсрэг арга юм. Энэ арга нь программ хангамжийн модуль тус бүрийн зардлыг тооцож гаргаад улмаар тэдгээрийг нэгтгэж, нийт зардлыг гаргадаг. Системийн бүрэлдэхүүн хэсэг, дэд систем, модуль тус бүрийн үнэлгээг нарийвчлан тооцдог онцлогтой.
- Price-to-Win арга. Программ хангамжийн функциональ хамаарлаас илүүтэйгээр захиалагчийн төсөв дээр түлхүү анхаардаг арга. Программ хангамжийн нийт зардлыг хөгжүүлэлтийн үндсэн санал дээр харилцан тохиролцоо замаар гаргадаг. Өөрөөр хэлбэл, программ хангамжийн хөгжүүлэлтийн зардлыг төслийн төсвөөр хязгаарлагддаг.

Б. Алгоритмын аргууд

Алгоритмын аргууд нь математик болоод тооцооллын үндсэн дээр зардлыг тооцоолж гаргадаг. Зардал тооцоолох томъёо нь судалгааны үндэслэлтэй байж, оролт нь шугаман код, функцийн цэг, зардлын эх үүсвэр болох эрсдэлийн тооцоолол, программчлалын хэл, аргачлалын дизайн зэргээс бүрддэг. Энэ төрлийн загваруудаас COCOMO, Putnam, функцийн цэг загвар, SEER-SEM загвар өргөн хэрэглэгддэг. Тухайлбал:

COCOMO Ажлын өртөг загвар

Barry Boehm нарын гаргасан зардлын тооцоо, төслийн хөгжүүлэлтийн хугацааг зэрэг тооцдог түгээмэл хэрэглэгддэг арга бол COCOMO (ажлын өртөг загвар) юм. Энэ аргад ашигладаг параметр, томъёо нь өмнө нь хэрэгжүүлсэн программ хангамжийн төсөлтэй уялдаатай байдаг. Программын кодын хэмжээг KLOC буюу мянган кодын шугамаар, харин хөдөлмөр/ажлыг хүн, сараар илэрхийлдэг. Boehm COCOMO загварын 3 төрлийг санал болгосон байдаг. Үүнд:

- Энгийн COCOMO (анхны загвар). Томъёо нь: Хөдөлмөр = $a \cdot (KLOC)^b$ Үүнд: kloc нь кодын хэмжээ, a болон b нь тогтмол хэмжигдэхүүн болно. a болон b утгын онцлогоос (төслийн төрөл, үндсэн эсэх, хагас тусдаа эсвэл нэгдсэн гэх мэт) хамаарна.
- Дундын COCOMO. Томъёо нь: Хөдөлмөр = $a \cdot (KLOC)^b \cdot EAF$ (1), энд хүчин чармайлал тохируулах хүчин зүйл нь EAF болно.

- Нарийвчилсан COCOMO. Энэ нь дэд систем тус бүр дээр ажилладаг болон нэгэн төрлийн дэд системүүдээр ерөнхий системдээ дэмжлэг болдог.

COCOMO II арга

COCOMO ажлын өртөг загварууд систем болон программын шаардлага нь тогтмол, эсвэл урьдчилан тодорхойлогдоно гэж тооцоолдог. Гэвч мэдээллийн систем, программ хангамжийн төсөл үргэлж дээрх шиг байдаггүй. Уг загварын онцлогоос үүдэн 1990 онд өгөгдлийн багцын өргөн хүрээтэй COCOMO II загвар бий болсон. Энэ аргад шугаман код, функцийн цэг, объектын цэгүүдийг оролт болгон ашиглаж, өмнөх COCOMO загварын зардлын эх үүсвэр болох хүч чармайлтын үржүүлэлтийг өөрчилсөн. Гарсан үр дүн нь хэмжээ болон хүч чармайлтын тооцоогоор илэрхийлэгддэг бөгөөд түүнд үндэслэн төслийн хуваарийг боловсруулдаг.

Putnam загвар

Энэ загвар нь олон программын төсөл судалдаг бөгөөд хүн хүчийн хуваарилалтаас хамаардаг.

$$S = E \cdot \text{хөдөлмөр}^{1/3} \cdot \text{td}^{4/3} \quad (2)$$

Үүнд:

S нь программын хэмжээг мөр тус бүрийг кодоор илэрхийлдэг (LOC/нэг мөр код) Тухайн орчны хүчин зүйл нь E хөгжлийн чадавхийг илэрхийлдэг. td нь программ бэлэн болох хугацааг илэрхийлэх ба хөдөлмөр нь хүн жилээр тооцогддог. Үүнээс гадна Putnam загварт нэг хамаарал олсон юм.

$$\text{Хөдөлмөр} = D0 \cdot t_d^3 \quad (3)$$

Энд хүн хүч илэрхийлдэг $D0$ нь шинэ болон дахин боловсруулсан программд 8-27 хооронд хэлбэлздэг. Putnam загвар дээр үндэслэдэг SLIM гэх арга хэрэгсэл нь зардлын тооцоо болон хүн хүчний хуваарь гаргахад ашиглагддаг.

Функцийн цэг дээр үндэслэсэн шинжилгээ

Albrecht функцийн цэгийн шинжилгээг программ хангамжийн төслийн үйл ажиллагааг хэмжих зорилгоор хөгжүүлжээ [9]. Энэхүү аргачлал нь программын хэмжээг тодорхойлж, дотоод логикийн файл, гадаад интерфэйс, гадаад оролт ба гаралтыг хэмждэг. Жишээлбэл, $estimas$ ба $SPQR/20$ бол функцийн цэгийн судалгаан дээр үндэслэсэн арга болно.

2.1. Аргуудын харьцуулалт

Бид судалгааны ажлын хүрээнд дээр дурдсан аргуудын давуу ба сул талыг дараах байдлаар тодорхойлж байна. Хүснэгт 1.

№	Арга	Давуу тал	Сул тал
1	Адилтгалд суурилсан үнэлгээ	Өмнөх төслүүдийн өгөгдөл болон үнийн дүнгээс хамаарна. Үнэлгээчийн чадвараас зардлын тооцооллын оновчтой байдал хамаарна.	Шинжээчийг чухал хэрэгтэй мэдээллээр хангах төслийг тодорхой шинж чанараар тодорхойлох шаардлагатай байдаг. Төсөл бүр дээр ашиглах боломжгүй.
2	Мэргэжилтний үнэлэмжийн арга	Мэргэжилтний таамаглалаас хамаарч технологи, архитектур, программчлалын хэлийг ашиглана.	Баримтжуулахад төвөгтэй. Үнэлгээчин, шинжээчид бүрэн найдаж болохгүй.
3	Дээрээс-доош тооцооллын арга	Төслийн талаар дэлгэрэнгүй мэдээлэл шаарддаггүй.	Жижиг хэмжээний асуудал, системийн зардлын нэмэгдэл, нийлмэл асуудлыг харгалзан үздэггүй.
4	Доороос-дээш тооцооллын арга	Тооцооллын найдвартай байдал өндөр.	Баримт бичгийн боловсруулалттай холбоотой зардал зэрэг системийн түвшний үйл ажиллагаанд анхаардаггүй.
5	SOCOMO загвар	Тооцоолоход хялбар.	Программ хангамжийн хөгжүүлэлтийн эхэн үед хэрэглэдэг учир тооцооллын алдаа гарах магадлалтай.
6	SOCOMO II	Үйлдвэрийн стандарт загвар. Үйл явц нь тодорхой, хялбар тохируулагддаг.	Жижиг хэмжээний төслийн үргэлжлэх хугацааны тооцоолол үндэслэлгүй гарах нь бий.
7	Putnam загвар	Цаг хугацаа, хэмжээ гэсэн хувьсах хоёр хүчин зүйлд үндэслэгддэг.	Программ хангамжийг хөгжүүлэх амьдралын циклын бусад асуудлыг авч үздэггүй.
8	Функцийн цэгийн шинжилгээ	Программчлалын хэл, хэрэгсэл, хэрэгжүүлэх аргаас хамааралгүй. Хөгжүүлэлтийн зардлыг программ хангамжийн хөгжүүлэлтийн эхэн үе шатанд тооцоолж болно.	Цаг хугацаа их шаарддаг, гар ажиллагаа ихтэй. Функцийн цэгийн хэрэглээг баримжаалахад туршлага шаарддаг.

Программ хангамжийн зардлыг тооцоолох хамгийн сайн, шалгарсан нэг арга гэж байдаггүй. Иймээс нөхцөл байдалд нийцүүлэн тооцооллын аргыг оновчтой сонгон ашиглах нь чухал. Практикт тооцооллын хэд хэдэн аргыг хослуулан ашиглах нь зардлын тооцоог илүү оновчтой гаргадаг. Бид ажилдаа тооцоолоход хялбар арай илүү өндөр түвшинд тооцолдог учраас SOCOMO аргыг сонгон хэрэглэсэн.

2.2 SOCOMO Ажлын өртөг загвар

Өртгийн тооцоо, төслийн хөгжүүлэлтийн хугацааг зэрэг тооцдог SOCOMO аргад ашиглах параметр, томъёо нь өмнө хэрэгжүүлсэн программ хангамжийн төсөлтэй уялдаатай байдаг. Энгийн регрессийн томъёог ашиглан параметруудийг өмнөх төслийн өгөгдлөөс гаргаж, хэрэгжиж буй төслийн шинж чанарт үндэслэн тохируулдаг. SOCOMO арга нь үндсэн, дундын, нарийвчилсан гэсэн хэлбэртэй бөгөөд программ хангамжийн хөгжүүлэлтэд 3 горимоор тодорхойлдог (Robert E. Park, 1994).

Хүснэгт 1. SOCOMO аргын хөгжүүлэлтийн горим

Шинж чанар	Органик	Хагас таслалтын горим	Угсармал
Бүтээгдэхүүн, зорилтуудын талаарх ойлголт	Нарийн нягт	Их	Ерөнхий
Хамааралтай программ хангамжийн системтэй ажиллах туршлага	Маш өндөр	Их	Дунд зэргийн
Урьдчилан тодорхойлсон хэрэглэгчийн шаардлагатай программ хангамжийн нийцтэй байдал	Хялбар	Их	Бүрэн
Гадаад интерфэйсийн техникийн нөхцөлтэй программ хангамжийн нийцтэй байдал	Хялбар	Их	Бүрэн

Үндсэн SOCOMO нь программын функцийн хэмжээнээс хамааран түүнийг хөгжүүлэх хүчин чармайлтыг тооцдог. Хүчин чармайлтын хэмжээг кодын мөрийн хэмжээ (KDSI)-ээр тооцоолдог.

$$\begin{aligned} \text{Хүчин чармайлт(хүн, сар)} &= a_i \times (KLOC)^{b_i} \times EAF \\ \text{Хөгжүүлэлтийн хугацаа (сар)} &= c_i \times (\text{Хүчин чармайлт})^{d_i} \\ \text{Бүтээмж} &= KDSI/(\text{Хүчин чармайлт}) \\ \text{Боловсон хүчний дундаж хэмжээ} &= \\ (\text{Хүчин чармайлт})/(\text{Хөгжүүлэлтийн хугацаа}) \end{aligned} \quad [1]$$

Үүнд:

KDSI (Kilo Delivered Source Instructions)-Кодын мөрийн хэмжээ KLOC(thousands of lines code) - мянган кодын мөрийн хэмжээ буюу программын эх кодыг хэмжих хэмжээс EAF (effort adjustment factor - программ хангамжийг бүтээх өртөгт нөлөөлөх хүчин зүйлсийн үржвэрээр тодорхойлогдсон хүчин чармайлт тохируулах хүчин зүйл COCOMO аргын тогтмол утгуудыг Хүснэгт 2 дээр харуулав.

Хүснэгт 2. Үндсэн COCOMO аргын тогтмол утгууд

Горим	a _b	b _b	c _b	d _b
Органик	2.40	1.05	2.50	0.38
Хагас таслалтын горим	3.00	1.12	2.50	0.35
Угсармал	3.60	1.20	2.50	0.32

2.3. Ажлын өртөг загвар COCOMO II арга

2000 онд гарсан COCOMO II аргад кодын мөрийн хэмжээ, программ хангамжийг бүтээх өртөгт нөлөөлөх хүчин зүйл, функцийн цэгийг ашигладаг.

$$\begin{aligned} \text{Хүчин чармайлт(хүн, сар)} &= 2.94 \times \text{Size}^E \times \prod EM_i \\ E &= 0.91 + 0.01 \times \sum SF_j \\ \text{Хөгжүүлэлтийн хугацаа} &= 3.67 \times (\text{Хүчин чармайлт})^F \\ SF &= 0.28 + 0.2 \times (E - 0.91) \end{aligned} \quad [2]$$

Үүнд:

Size- Функцийн цэг эсвэл кодын мөрийн хэмжээ

E- Хүчин чармайлтын хэмжээ

SF- Хураангуйлсан хүчин зүйл (Scale factor)

EM- хүчин чармайлтын үржвэрүүд



Зураг 3. COCOMO II бүрэлдэхүүн хэсгийн төрлүүд

Зурагт 3 дээр COCOMO II аргуудын тооцооллын үндэслэл, тэдгээрийн хэрэглээг ялган харуулав. Програмын бүрэлдэхүүн загварыг объектын цэгийн хэмжээг үндэслэн тооцоологддог бол өмнөх үеийн загварт кодын мөрийн хэмжээ тодорхойгүй тул функцийн цэгийн шинжилгээнд үндэслэж тооцоолдог байсан (Parthasarathy, 2007). Дахин ашиглалтын загварын тухайд түүхчилсэн өгөгдөлд үндэслэж дахин ашиглагдах кодын мөрийн хэмжээгээр тооцоолох ба архитектурын дараах загвар нь түүхчилсэн өгөгдөлд үндэслэж кодын мөрийн хэмжээгээр тооцоолдог.

COCOMO загварт өртөгт нөлөөлөх хүчин зүйлсийг дараах 4 бүлэгт хуваадаг. Программ хангамжийг бүтээх өртөгт нөлөөлөх дараах хүчин зүйлийг авч үздэг. Үүнд:

ACAP: Шинжээчийн ур чадвар

PCAP: Программ зохиогчийн ур чадвар

RELY: Программ хангамжийн найдвартай байдал

DATA: Өгөгдлийн сангийн хэмжээ

CPLX: Бүтээгдэхүүний төвөгтэй байдал

TIME: Гүйцэтгэх хугацааны хязгаарлалт

STOR: Үндсэн санах ойн үзүүлэлтүүд

VIRT: Техник хангамж, программ хангамжийн хүчин чадал

TURN: Компьютер шинэчлэх хугацаа

AEXP: Программ хангамжтай ажиллаж байсан туршилага

VEXP: Виртуаль машинтай ажиллах туршилага

LEXP: Хэлний туршилага

MOD: Шинээр гарч байгаа программ хангамжийн технологитой ажиллах туршилага

TOOL: Программ хангамжийн хэрэгсэл ашиглах чадвар

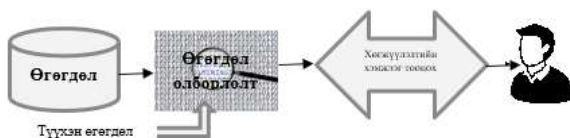
SCED: Хөгжүүлэлтийн цагийн хуваарь

Хүчин зүйл	Маш бага	Бага	Дундаж	Их	Маш их	Туйлын их
ACAP	1.46	1.19	1	0.86	0.71	
PCAP	1.42	1.17	1	0.86	0.7	
AEXP	1.29	1.13	1	0.91	0.82	
MODP	1.24	1.1	1	0.91	0.82	
TOOL	1.24	1.1	1	0.91	0.83	
VEXP	1.21	1.1	1	0.9		
LEXP	1.14	1.07	1	0.95		
SCED	1.23	1.08	1	1.04	1.1	
STOR			1	1.06	1.21	1.56
DATA		0.94	1	1.08	1.16	
TIME			1	1.11	1.3	1.66
TURN		0.87	1	1.07	1.15	
VIRT		0.87	1	1.15	1.3	
RELY	0.75	0.88	1	1.15	1.4	
CPLX	0.7	0.85	1	1.15	1.3	1.65

3. Программ хангамжийг бүтээх өртгийг тооцоолох шийдэл

Өгөгдөл олборлолт нь их хэмжээний багц өгөгдөлд дүн шинжилгээ хийж, өгөгдлийн утгыг компьютерийн тусламжтайгаар олборлох болон төрөл бүрийн хэтийн төлөвлөгөөнд хэрэгцээтэй мэдээлэлд дүн шинжилгээ, дүгнэлт хийх үйл явц юм (Ц.Лхамролом, 2015). Үүнийг өгөгдлийн сангийн их хэмжээний өгөгдөлд олон талаас дүн шинжилгээ хийж, тэдгээрийн далд, өмнө нь мэдэгдээгүй, ашигтай мэдээллийн холбоо хамаарлыг илрүүлж ашигтай загвар үүсгэн мэдлэг бий болгох үйл явц хэмээн тодорхойлж болно. Мэдлэгийн нээлтийн үйл ажиллагааны үе шатуудыг авч үзвэл:

- Олон тооны өгөгдлийн эх үүсвэрээс өгөгдлийг нэгтгэж, өгөгдлийн агуулах үүсгэнэ.
- Өгөгдлийг урьдчилан боловсруулж стандарт форматад оруулна.
- Өгөгдлийн уурхайн арга техникийг ашиглан загвар үүсгэнэ.
- Энэхүү үүсгэсэн загварыг ашиглан шинэ, ашигтай мэдлэгийг олж авна (Ш.Цэнд-Аюуш, 2015).
- Өгөгдөл олборлолтын аргыг программ хангамжийн үйл явцын чанарыг сайжруулах, хүчин чармайлтыг нарийвчлалтайгаар тооцоолоход COCOMO аргатай хослуулан программ хангамжийг хөгжүүлсэн түүхэн өгөгдлийг ашиглан өгөгдөл олборлолтын алгоритмыг хөгжүүлэлтийг хэмжээг тооцох боломжтой (Snehal A.Deshmukh, 2016).



Зураг 4. Өгөгдөл олборлолтын аргыг ашиглан хөгжүүлэлтийн хэмжээг тооцох ерөнхий загвар

3.1. Программ хангамжийг бүтээх өртгийг өгөгдөл олборлолтын аргатай хослуулсан ерөнхий загвар

COCOMO аргаар хүчин чармайлтыг тодорхойлоход кодын мөрийн хэмжээ шаарддаг. Хэрэв кодын мөрийн хэмжээ тодорхойгүй бол функцийн цэгийн шинжилгээний аргаар хүчин чармайлтыг тодорхойлох боломжтой. Иймд бид өртөг тооцоход COCOMO болон функцийн цэгийн шинжилгээний аргуудыг дэлгэрүүлж судалсан. Өгөгдөл олборлолтын аргыг COCOMO өртөг

тооцох аргын өртөгт нөлөөлөх хүчин зүйлс дээр шинжилгээ хийж, холбоо хамаарлыг илрүүлэхэд ашигласан (Koskenkyla, 29-30 April 2015).



Зураг 5. Программ хангамжийг бүтээх өртөг тооцох загвар

Түүхчилсэн бодит өгөгдөл дээр өгөгдөл олборлолтын алгоритмын тусламжтай загвар үүсгэсэн. Уг загвартаа одоогийн оролтын өгөгдлөөр шинжилгээ хийж, үр дүнд өртөг тооцох аргаар программ хангамжийг бүтээх өртгийг тооцоолох зорилго тавьсан. Өгөгдөл олборлолтыг ашиглан төрөл бүрийн хэтийн төлөвлөгөөнд хэрэгцээтэй мэдээлэлд дүн шинжилгээ, дүгнэлт хийдэг. Ийм төрлийн мэдээллийг орлого нэмэгдүүлэх, зардлыг бууруулахад ашиглаж болно. Программ хангамжийг бүтээх өртгийг тооцоолдог түгээмэл хэрэглэгддэг хүчин зүйлсээс хамгийн чухал хүчин зүйлсийг өгөгдлийн уурхайн Weka программ ашиглан харьцуулахыг зорьсон. Программ хангамжийг бүтээх өртөгт нөлөөлөх хүчин зүйлсийг программ хангамжийн төслийг дуусгахад шаардагдах хүчин чармайлтыг тодорхойлоход ашигладаг (Shivangi Shekhar, May 2016). Эдгээр хүчин зүйлс нь өртгийн тооцооны загварын үндэс болдог. Түүхчилсэн өгөгдөл болох программ хангамжийг бүтээх өртгийг тооцох хүчин зүйлсийг шинэ төсөлтэй харьцуулж тооцоолдог. Судалгааны гол зорилго нь өгөгдөл олборлолтын технологийн тусламжтайгаар программ хангамжийг бүтээх өртгийн тооцооллын үр ашгийг нэмэгдүүлэхэд оршино. Туршилтын өгөгдлийг бэлдэхэд COCOMO арга дээр тооцоологдсон их өгөгдөл шаардлагатай байсан тул бид PROMISE Data сангаас (Available: <http://promise.site.uottawa.ca/SERepository>) COCOMO арга дээр тооцоологдсон 60 төслийн багц өгөгдлийг ашигласан болно. PROMISE багц өгөгдөл нь программ хангамжийн инженерчлэлд давтагдаж болохуйц, шалгах, няцаалт өгөх, боловсруулж болох урьдчилан таамагласан загварыг олон нийтэд нээлттэйгээр тохируулсан мэдээллийн сан юм. Өгөгдлийн файл нь arff ба .csv форматтай, Weka болон бусад өгөгдөл олборлолтын программд шууд ашиглах боломжтой. Уг өртөгт нөлөөлөх хүчин зүйлс нь өгөгдсөн 300 гаруй багц өгөгдлийн 60 хувийг судалгааны өгөгдөл, үлдсэн 40 хувийг туршилтын өгөгдөл болгон ашигласан.

4. Программ хангамжийг бүтээх зардал өгөгдөл олборлолтыг ашигласан туршилт

Алхам 1. Өгөгдөл олборлолтын к-ын дунджаар бүлэглэх

Программ хангамжийг бүтээх өртгийг таамаглахын тулд туршилтын өгөгдлийг к-ын

Хүснэгт 4. Туршилтын өгөгдлийг к-ын дунджаар бүлэглэх аргаар 5 бүлэгт хуваасан ангиллын үр дүн

Хүчин зүйлс	Бүлэг -1	Бүлэг -2	Бүлэг -3	Бүлэг -4	Бүлэг -5
Программ хангамжийн найдвартай байдал /RELY/	Дунд, Их	Дунд	Бага, Дунд	Их	Дунд
Өгөгдлийн сангийн хэмжээ /DATA/	Бага	Дунд	Дунд, Их	Дунд	Маш их
Бүтээгдэхүүний төвөгтэй байдал /CPLX/	Их	Их	Дунд, Маш их	Их	Их
Гүйцэтгэх хугацааны хязгаарлалт /TIME/	Дунд	Дунд	Дунд	Их	Маш их
Санах ойд эзлэх хэмжээ /STOR/	Дунд	Дунд	Дунд	Их	Маш их
Техник ба программ хангамжийн хүчин чадал /VIRT/	Бага	Дунд	Бага	Бага	Бага
Компьютерийн шинэчлэх хугацаа /TURN/	Бага	Дунд	Бага, Дунд	Их	Их
Программ хангамжтай ажиллах туршлага /AEXP/	Дунд-Маш их	Дунд, Их	Дунд-Маш их	Их	Их
Программ зохиогчийн чадвар /PCAP/	Дунд-Маш их	Дунд, Их	Маш их	Дунд	Дунд
Шинжээчийн чадвар /ACAP/	Дунд, Их	Дунд, Их	Их	Дунд	Маш их
Виртуаль машинтай ажиллах чадвар /VEXP/	Дунд	Дунд	Бага, Дунд	Дунд	Бага
Хэлний туршлага /LEXP/	Их	Дунд, Их	Дунд, Их	Дунд	Их
Шинээр гарч байгаа программ хангамж, технологитой ажиллах туршлага /MODP/	Дунд, Их	Дунд, Их	Бага, Их	Бага	Маш их
Программ хангамжийн хэрэгсэл ашиглах чадвар /TOOL/	Дунд	Дунд	Бага, Дунд	Маш их	Маш их
Хөгжүүлэлтийн цагийн хуваарь /SCED/	Бага, Дунд	Дунд	Бага-Их	Дунд	Бага
Программын мөрийн хэмжээ /LOC/	5.5-302	8-90	70-423	21-219	12.8-48.5

дунджаар бүлэглэх алгоритмаар төслийн өртөгт нөлөөлөх шинж чанараар ангилж үзсэн. СОСОМО өртөг тооцоолох арга нь органик, хагас таслалтын, угсармал гэсэн 3 горимтой байдаг тул туршилтын өгөгдлийг 5 бүлэгт хувааж үзэхэд Хүснэгт 4-т харуулсан үр дүн гарсан.

Туршилтын өгөгдлийг 5 бүлэгт ангилснаар өртөгт нөлөөлөх хүчин зүйлсийг илүү нарийвчлалтай тодруулсан. Үр дүнгээс хамаарч бүлэг тус бүрийн СОСОМО 81 аргын горимыг дараах байдлаар тодорхойлсон (Snehal A.Deshmukh, 2016) (Хүснэгт 6).

Хүснэгт 6. Өртөгт нөлөөлөх хүчин зүйлсийн шинжээр тооцоолсон байдал

	Өртөгт нөлөөлөх хүчин зүйлсийн шинжээр	
1	Бүлэг-1	органик горимд
2	Бүлэг-2	хагас таслалтын горимд
3	Бүлэг-3	хагас таслалтын горимд
4	Бүлэг-4	угсармал горимд
5	Бүлэг-5	угсармал горимд

Алхам 2. Бодит болон тооцооллын үр дүнгийн харьцуулалт

Туршилтын өгөгдөлд байгаа өртгийн хэмжээ нь бодит өртгийн хэмжээ бөгөөд СОСОМО аргаар тооцоолсон өртгийг органик, хагас таслалтын, угсармал горимд тооцоолж гаргасан. Алхам 1-д гаргасан горимын тодорхойлолтыг бататгахын тулд бүх туршилтын өгөгдлийг, 3 горимоор тооцооллоо (Sumeet Kaur Sehira, 2014). Үүнд:

- Хүснэгт 21. Бүлэг-1 өгөгдлийг органик горимын томьёогоор тооцоолж, бодит өртөгтэй харьцуулсан зөрүү
- Хүснэгт 22. Бүлэг-2 өгөгдлийг өртгийг хагас таслалтын горимоор тооцоолж, бодит өртөгтэй харьцуулсан зөрүү
- Хүснэгт 23. Бүлэг-3 өгөгдлийг хагас таслалтын горимын томьёогоор тооцоолж, бодит өртөгтэй харьцуулсан зөрүү
- Хүснэгт 24. Бүлэг-4 өгөгдлийг угсармал горимын томьёогоор тооцоолж, бодит өртөгтэй харьцуулсан зөрүү
- Хүснэгт 25. Бүлэг-5 өгөгдлийг угсармал горимын томьёогоор тооцоолж, бодит өртөгтэй харьцуулсан зөрүү.

Алхам 3. Бодит болон тооцоолсон өртгийн алдааны зөрүү

Тооцоолсон утгуудыг ангилсан горимуудын хувьд тооцооллын зөрүүг хувиар илэрхийлсэн.

Хүснэгт 7. Бүлэг-1 өгөгдлийг органик горимоор тооцоолсон өртөг ба бодит өртгийн зөрүү, дүн

№	Бодит	Органик	Хагас таслалтын	Угсармал	Зөрүү утга
2	324	369.12	491.44	684.44	13%-иар их
5	117.6	86.02	101.31	122.71	26.8%-иар бага
8	60	65.31	75.51	89.57	8.85%-иар их
9	360	154.58	200.04	269.87	57.06%-иар бага
10	60	44.91	50.64	58.38	25.15%-иар бага
11	48	25.91	29.36	34.03	46.025%-иар бага
12	25.2	31.54	34.74	38.99	25%-иар их
13	60	115.56	138.08	170.06	92.6%-иар их
16	72	28.53	32.98	39.12	60.3%-иар бага
17	48	35.05	40.52	48.06	28%-иар бага
20	360	323.16	418.2	564.19	10.2%-иар бага
28	480	413.6	540.84	738.34	13.8%-иар бага
29	352.8	232.42	292.47	382.12	34.12%-иар бага
31	300	232.42	292.47	382.12	22.53%-иар бага
33	18	18.45	19.61	21.12	2.5%-иар их
34	400	191.33	243.55	322.43	52.1%-иар бага
35	2400	1174.03	1641.54	2419.3	48.9%-иар бага

Нийт өгөгдлөөс амжилттай (25% дотор), 25%-иас дээш зөрүүтэй тооцоологдсон.

Хүснэгт 8. Бүлэг-2 өгөгдлийг хагас таслалтын горимоор тооцоолсон өртөг ба бодит өртгийн зөрүү, дүн

№	Бодит	Органик	Хагас таслалтын	Угсармал	Зөрүү
3	60	63.71	76.12	93.75	26.9%-иар их
14	450	280.61	360.48	482.23	19.9%-иар бага
15	210	150.96	182.56	227.95	13.1%-иар бага
18	48	28.1	30.95	34.73	35.5%-иар бага
22	252	159.88	196.54	250.03	22%-иар бага
26	42	32.67	35.42	39.04	15.7%-иар бага

Нийт 6 өгөгдлөөс 4 нь амжилттай (25% дотор), 2 өгөгдөл нь 25%-иас дээш зөрүүтэй тооцоологдсон.

Хүснэгт 9. Бүлэг-3 өгөгдлийг хагас таслалтын горимоор тооцоолсон өртөг ба бодит өртгийн зөрүү, дүн

№	Бодит	Органик	Хагас таслалтын	Угсармал	Зөрүү
0	2300	798.93	1143.73	1731.68	50.2%-иар бага
19	278	220.21	277.95	364.43	0.1%-иар бага
23	420	393.65	532.84	756.72	26.8%-иар их

Нийт 3 өгөгдлийн 1 нь амжилттай (25% дотор), 2 нь 25%-иас дээш зөрүүтэй байсан.

Хүснэгт 10. Бүлэг-4 өгөгдлийг хагас таслалтын горимоор тооцоолсон өртөг ба бодит өртгийн зөрүү, дүн

Тох №	Бодит	Органик	Хагас таслалтын	Угсармал	Зөрүү
4	2120	1104.22	1509.59	2168.36	2.2%-иар их
6	370	234.16	288.67	368.44	0.4%-иар бага
7	107	146.63	170.12	202.57	89%-иар их
24	750	465.4	602.7	813.44	8.4%-иар их

Нийт 4 өгөгдлөөс 3 нь амжилттай (25% дотор), 1 нь 25%-иас дээш зөрүүтэй тооцоологдсон.

Хүснэгт 11. Бүлэг-5 өгөгдлийг хагас таслалтын горимоор тооцоолсон өртөг ба бодит өртгийн зөрүү, дүн

Тох №	Бодит	Органик	Хагас таслалтын	Угсармал	Зөрүү
1	82	57.02	64.9	75.62	7.8%-иар бага
21	170	121.93	146.01	180.26	6%-иар их
25	239	184.66	227.33	289.68	21.2%-иар их
27	70	53.28	60.38	69.98	0.1%-иар бага
30	192	133.59	160.95	200.1	4.2%-иар их
32	62	45.85	51.43	58.94	4.9%-иар бага

Нийт 6 өгөгдөл бүгд амжилттай (25% дотор) тооцоологдсон.

Туршилтын үр дүнд бүлэг-1 өгөгдлийг 40 хувь, бүлэг-2 өгөгдлийг 88.8 хувь, бүлэг-3 өгөгдлийг 100 хувь, бүлэг-4 өгөгдлийг 100 хувь, бүлэг-5 өгөгдлийг 66.6 хувийн амжилттай таамагласан. Нийт таамаглалын дундаж нь 79.1 хувийн амжилттай байсан. Амжилтгүй таамагласан шалтгааныг илрүүлэхээр өртөгт нөлөөлөх хүчин зүйл дээр шинжилгээ хийсэн боловч тодорхой үр дүн гараагүй болно.

Дүгнэлт

Судалгааны ажлын хүрээнд программ хангамжийн хөгжүүлэлтийн хэмжээг урьдчилан тооцоолж, өгөгдөл олборлолтын алгоритм ашиглан таамаглах зорилго тавьж, дараах аргуудыг сонгон авсан. Үүнд:

- СОСОМО
- Өгөгдөл олборлолтын k-ын дунджаар бүлэглэх алгоритм
- Өгөгдөл олборлолтын Apriori алгоритм
- Өгөгдөл олборлолтын WEKA хэрэгсэл

Программ хангамжийн хөгжүүлэлтийн хэмжээг тооцоход шаардлагатай туршилтын өгөгдөлд СОСОМО аргаар тооцоологдсон бэлэн багц өгөгдлийг ашигласан. Ингэхдээ программ

хангамжийг бүтээх өртгийг таамаглахын тулд k-ын дунджаар бүлэглэсэн байсан судалгааны загварт туршилтын өгөгдлийг оруулж тохирох бүлгийг сонгосон. Эдгээр бүлэг бүрийг судалгааны загварт авч үзсэн горим тус бүрээр тооцож үр дүнг харьцуулж үзлээ. Гарсан үр дүнд өгөгдөл олборлолтын аргаар таамаглах хувилбарыг авч үзсэн.

Судалгааны хүрээнд тодорхойлсон энэхүү хувилбарыг ашиглан жижиг хэмжээний программ хангамжийн хөгжүүлэлтийн хэмжээг урьдчилсан байдлаар тооцоолж, үр дүнг таамаглах боломжтой хэмээн үзэж байна. Цаашид судалгааны ажлыг дараах байдлаар өргөтгөх боломжтой. Үүнд:

- Практикт хэрэгжиж байгаа болон хэрэгжсэн төслийн бодит өгөгдлийг цуглуулах замаар туршилтын өгөгдлийн сан үүсгэх,
- Программ хангамжийн хэмжээг тооцоолоход шаардлагатай авч үзэх хүчин зүйлсийг нэмж тодорхойлох,
- Тооцооллын үр дүнг таамаглах хувилбарыг боловсронгуй болгох.

Ном зүй

1. Benala T.R., M. R. (2014). Software Effort Estimation Using Data Mining Techniques. ICT and Critical Infrastructure: Proceedings of the 48th Annual Convention of Computer Society , vol 248. . India.
2. Capt Gregory E.Brown, U. E. (April 2017). An Investigation of Nonparametric Data Mining Techniques for Acquisition of Cost Estimationg Nonparametric Data Mining Techniques ., Vol24, №2-302-332.
3. Dr.Ulbert, Z. (2014). Software development processes and software quality assurance, "Chapter 13. University of Pannonia.
4. HeydarNonori, A. (2000). Function point analysis. Koskenkyla, J. (29-30 April 2015). Cost estimation in global software development. Barcelona, Spain: IEEE.
5. Parthasarathy, M. A. (2007). Practical Software Estimation: Function Point Methods for Insourced and Outsourced Projects. USA: Addison-Wesley Professiona.
6. Robert E. Park, W. B. (1994). Software Cost and Schedule Estimating (Ботд. Special Report, CMU/SEI-94-SR-3). USA: Carnegie Mellon University.
7. Shivangi Shekhar, U. K. (May 2016). Review of
8. Various Software Cost Estimation Techniques. Vol 141(International Journal of Computer Applications).
9. Snehal A.Deshmukh, P. S. (2016). Im-plementation of using classification Data Mining Techniques for Software Cost Estimation. In-ternational Journal on Recent and Innovation Trends in Computing and Communication , Vol4, issue 5;.
10. Sumeet Kaur Sehira, J. K. (2014). Analysis of Data Mining Techniques for Software Effort Estimation . 11th International Conference on Information Technology.
11. Ц.Лхамролом. (2015). Классын алдаанд өртөмтгий байдлыг объект хандлагат зохиомжийн хэмжигдэхүүн ашиглан таамаглах нь".
12. Ш.Цэнд-Аюуш. (2015). Өгөгдлийн уурхайг хөл бөмбөгийн тааварт ашиглах нь.

Using Data Mining in Software Effort Estimation

Uyanga S.^{1*}, Munkhtsetseg N.¹ and Naranmandal Ch.²

¹ Department of Information and Computer Science, School of Engineering and Applied Sciences,
National University of Mongolia, Ikh surguuliin gudamj-3, Ulaanbaatar-14201

² G-mobile LLC

*uyanga@seas.num.edu.mn

Received on 03.20.2018; revised on 09.17.2018; accepted on 09.20.2018

Abstract

Cost estimation and project management play a key role in rapid development for high-quality software products at low cost. Software cost estimation and cost benefit analysis, especially accurate effort estimation are very important for software engineering. In this paper, we suggest cost estimation model using data mining tools. We used weka tools for data mining and COCOMO tools for software effort estimation.

Key words: Software effort estimation, COCOMO, data mining, function point, cost drivers, Apriori algorithm, WEKA
