COURSE : FUNDAMENTALS OF COMPUTER SCIENCE

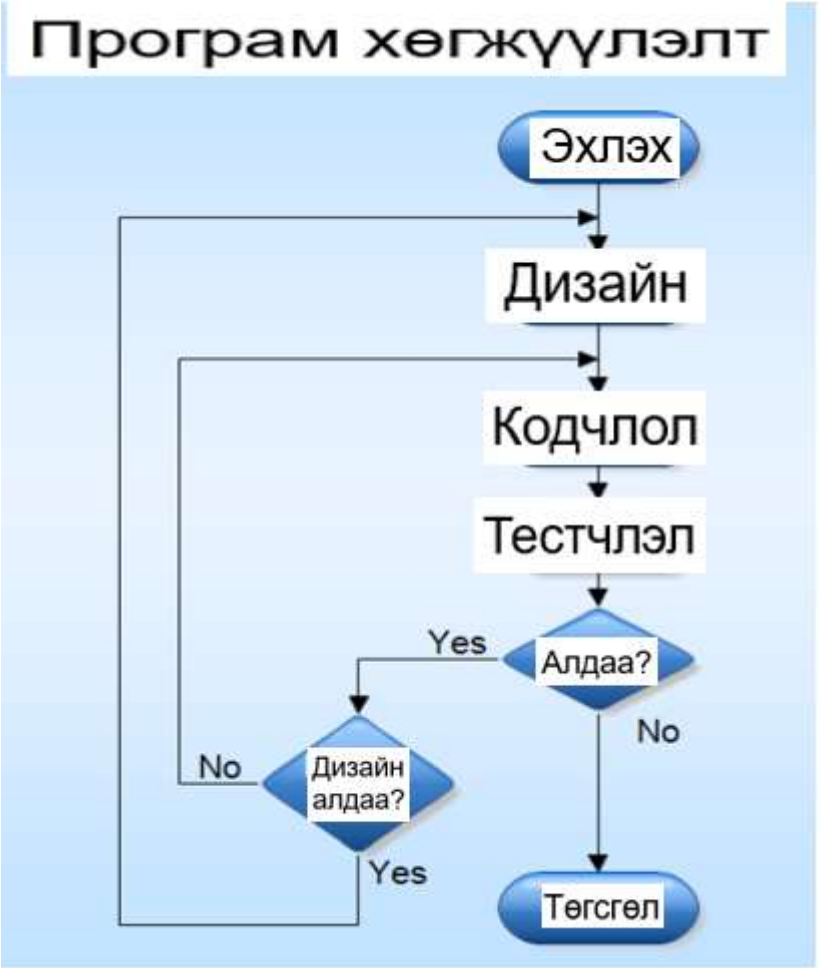
LECTURE 7: "SOFTWARE ENGINEERING, DEVELOPMENT PHASES"

Instructor:
PhD, Associate Professor
Tuyatsetseg Badarch

Оршил

Сэдвүүд:

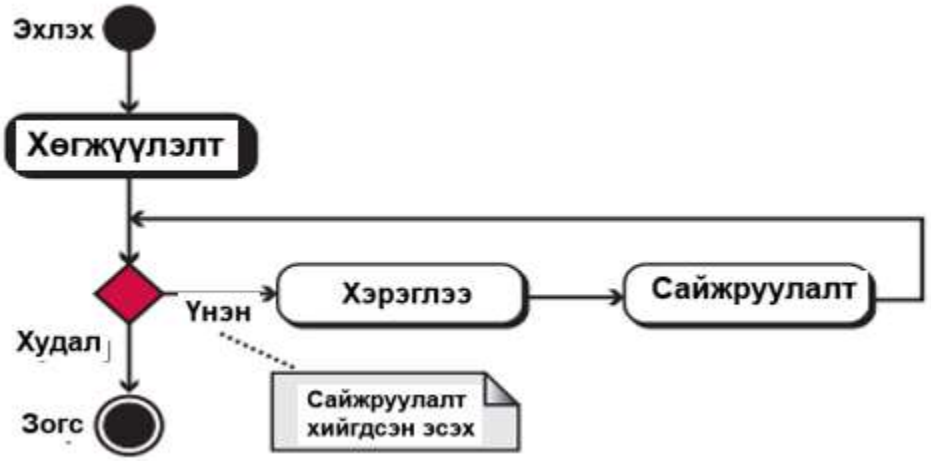
- Амьдралын цикл
- Анализ
- Дизайн
- Хөгжүүлэлт
- Тестлэл
- Докумэнчлэл



Зураг 1. Програмын хөгжүүлэлт

SOFTWARE LIFECYCLE

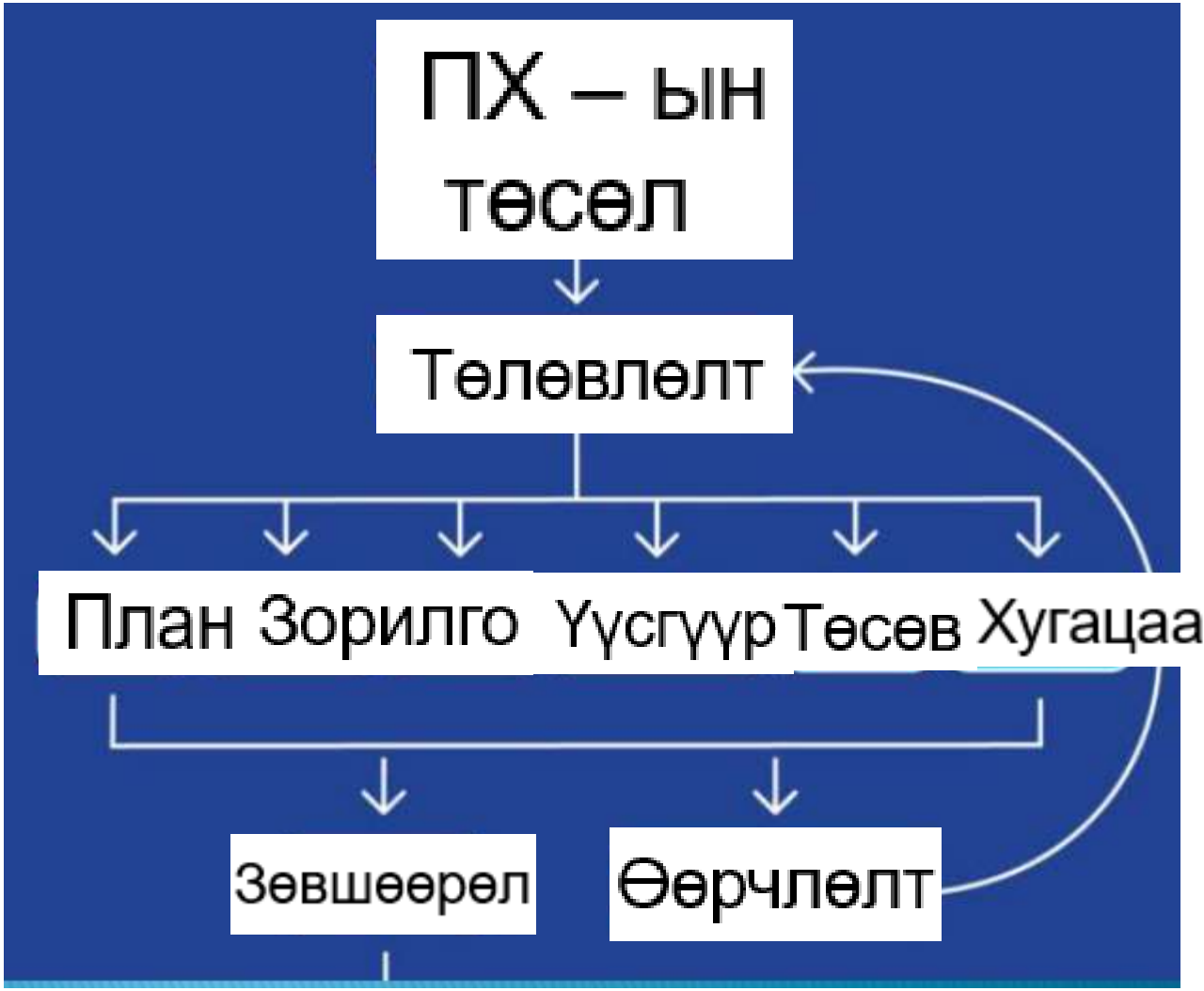
Програм хангамжийн инженерчлэлийн үндсэн ойлголт бол програм хангамжийн амьдралын мөчлөг (life cycle) юм. Програм хангамж гэх мэт бусад олон бүтээгдэхүүнүүд тодорхой үе шатууд дамжин хөгждөг.



Зураг 2. Програмын амьдралын цикл

Програм хөгжүүлэлтийн үндэс

Програм хангамжийн инженерчлэлийн үндсэн ойлголт бол програм хангамжийн амьдралын мөчлөг (life cycle) юм. Програм хангамж гэх мэт бусад олон бүтээгдэхүүнүүд тодорхой үе шатууд дамжин хөгждөг.



Зураг 3. Програмын төсөл

Програм хөгжүүлэлтийн үеийн өөрчлөлт

Change - Програм хангамжийг хөгжүүлэх явцад гарсан алдаанууд, түүнчлэн програм хангамжийн дизайныг зохицуулах дүрэм, хууль тогтоомжийн өөрчлөлт, эсвэл хэрэглэгчийн шаардлагаас үүдэн өөрчлөлт хийх шаардлагатай байдаг.

Хөгжүүлэлтийн явц: Waterfall model

Хөгжүүлэгчдийн процессын моделууд

Програм хангамж боловсруулах үйл явцын үеийн нэг чухал загварыг “Waterfall” гэж нэрлэдэг.

Энэ загварт хөгжлийн үйл явц зөвхөн нэг чиглэлд урсдаг. Өөрөөр хэлбэл, өмнөх үе шат дуусах хүртэл үе шатыг эхлүүлэх боломжгүй гэсэн үг юм.

Жишээ нь, давуу тал нь дараагийн үе шат эхлэхээс өмнө үе шат бүрийг дуусгах зарчимтай загвар болно.

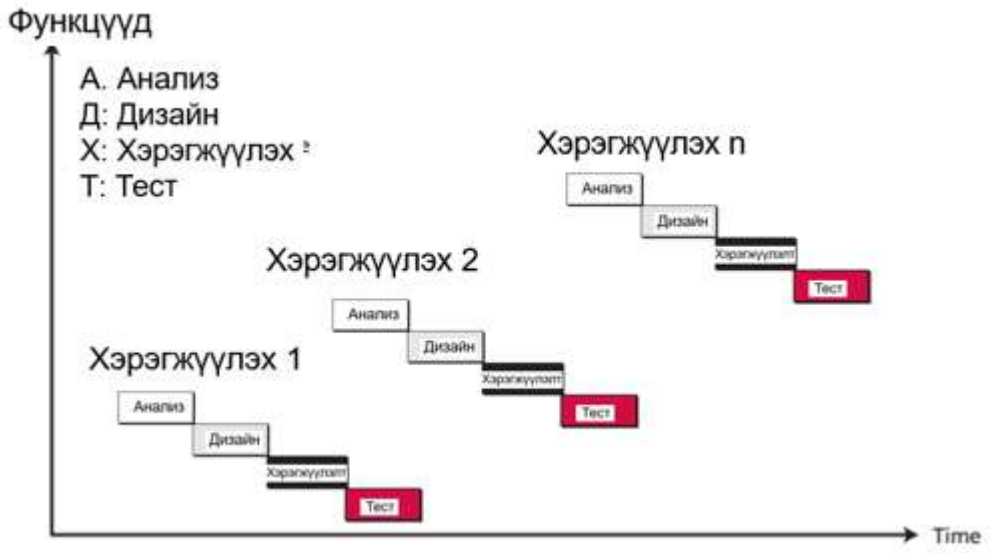
Гэхдээ “waterfall” загварын сул тал болох хэрэв үйл явцын зарим хэсэгт асуудал байгаа бол бүх үйл явцыг бүгдийг шалгах шаардлагатай. Энэ нь хугацаа, нөөц гэх мэт суурь зүйлүүдэд сөргөөр нөлөөлнө.

Зураг 4. The waterfall загвар

Хөгжүүлэлтийн явц: Incremental model

ПХ-ын хөгжүүлэлтийн турш улам төвөгтэй болох бүтэц нь хөгжүүлэлтийг хэд хэдэн үе шаттайгаар боловсруулахад хүргэдэг. Зурагт n үе шаттай болохыг үзүүлсэн.

Нэг. Хөгжүүлэгчид эхлээд бүхэл системийн хялбаршуулсан хувилбарыг боловсруулж дуусгадаг. Энэ хувилбар нь бүхэл системийг төлөөлдөг боловч дэлгэрэнгүй мэдээллийг оролцуулаагүй болно. Зурагт тус загварыг үзүүлэв.



Зураг 5. Incremental загвар

Foundations of Computer Science, Behrouz A. Forouzan, Fourth Edition, Cengage Learning EMEA, 2018, chapter 10

Хөгжүүлэлтийн явц: Анализ

ПХ-ийг хөгжлүүлэх үйл явц нь шинжилгээний үе шат (analysis phase)-аас эхэлдэг. Энэ үе шат нь програм хангамжийг хэрхэн яаж хийхийг зааж өгөхгүйгээр юу хийхийг харуулсан тодорхойлолтын баримт бичгийг тодорхойлдог.
Шинжилгээний үе шат нь програм хөгжүүлэлтийн явцад програмчлалын хэл эсвэл объект хандлагат хэл ашиглан хийгдсэн эсэхээс хамаарч хоёр тусдаа аргыг ашиглаж болно.

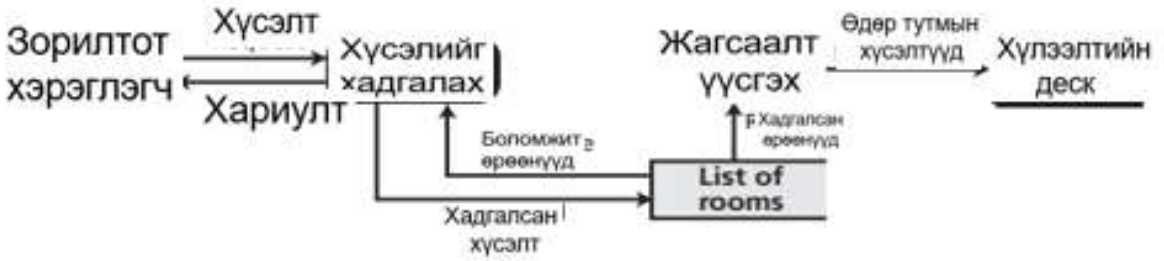
Процедурт чиглэсэн шинжилгээ

Процедурт чиглэсэн шинжилгээ (бүтэцлэгдсэн шинжилгээ –structured analysis) нь програм хангамжийг хэрэгжүүлэх үе шатанд процедурын хэлийг ашиглах тохиолдолд ашигладаг шинжилгээ юм. Энэ тохиолдолд техникийн үзүүлэлтүүд нь хэд хэдэн загварчлалын хэрэгслийг ашигладаг.

Хөгжүүлэлтийн явц: урсгалын диаграм

Өгөгдлийн урсгалын диаграм нь систем дэх өгөгдлийн шилжилт, үйл явцыг харуулдаг.

Зурагт үзүүлснээр, процессууд нь захиалгын файлыг ашиглан бэлэн эсэхийг шалгах мөн захиалгыг хүлээн авах эсвэл татгалзах үйлдлийг илэрхийлнэ. Захиалга хүлээн авагдсан тохиолдолд захиалгын файлд тэмдэглэгдэнэ.



Зураг 6 Өгөгдлийн урсгалын диаграм

Зурагт үзүүлснээр, дөрвөн тэмдэглэгээ хэрэглэгдэнэ: Тэгш өнцөгт тэмдэглэгээ нь мэдээллийн эх сурвалж эсвэл хүрэх төгсгөлийг, бөөрөнхий булантай тэгш өнцөгт нь процессыг, нээлттэй тэгш өнцөгт нь өгөгдөл хадгалагдаж байгаа байршил, сумнууд нь өгөгдлийн урсгалыг илэрхийлж байна.

Хөгжүүлэлтийн явц: Төвөлийн диаграм

Төлөвийн диаграммууд нь системийн үйл ажиллагааны төлөв өөрчлөгдөх үед ихэвчлэн ашиглагддаг.
Төлөвийн диаграммын жишээ болгон зорчигчийн лифтийн ажиллагааг авч үзлээ. Товчлуур дарахад цахилгаан шат хүссэн чиглэлдээ хөдөлдөг. Зорьсон газраа хүрэх хүртлээ өөр хүсэлтэд хариу өгөхгүй.

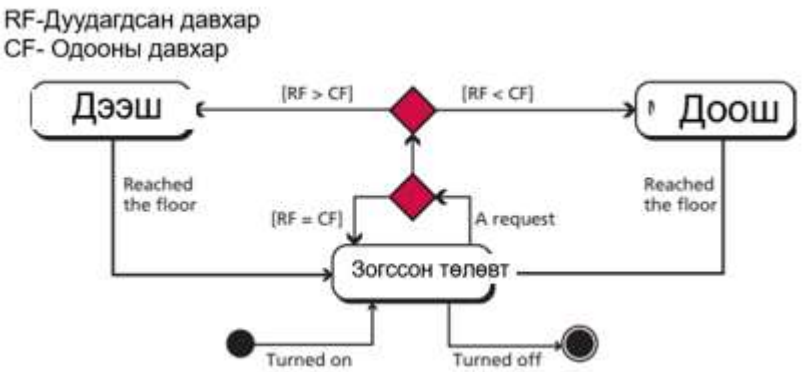
Зураг 7. Constructive Cost Model Cocomo /үнэ, үр дүн, цаг/

Хөгжүүлэлтийн явц: Төлөвийн диаграм

Зураг 10.5-д хуучин загварын лифтний төлөвийн диаграммыг үзүүлэв. Цахилгаан шат нь дээш, доошоо хөдлөх, зогссон буюу зогсоолтой төлөв гэсэн гурван төлвийн аль нэгэнд байж болно.

Эдгээр төлөв бүрийг төлөвийн диаграмм бөөрөнхий булантай тэгш өнцөгтөөр дүрсэлсэн байна. Лифт зогссон нөхцөлд хүсэлтийг хүлээн авдаг. Хэрэв хүссэн давхар нь одоогийн давхартай ижил байвал зогссон төлөвт үлдэнэ.

Хэрэв хүссэн давхар нь одоогийн давхраас дээш байвал лифт дээшээ чигтэй урсаж эхэлнэ. Хүссэн давхар нь хүссэн давхраас доогуур байвал цахилгаан шат доошоо урсаж эхэлнэ. Хөдөлсний дараа лифт хүссэн давхарт хүрэх хүртлээ нэг хөдөлгөөнт төлөвтэй.



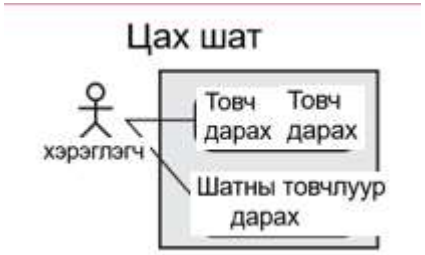
Зураг 8. Төлөвийн диагарм

Объект хандалтат шинжилгээ

Объект хандалтат шинжилгээ гэдэг нь объект хандалтат хэлийг ашигладаг шинжилгээний үйл явц юм.
Энэ тохиолдолд техникийн баримт бичигт хэд хэдэн програмчлалын хэрэгслийг ашиглаж болох гол хэдхэн зүйлсийг авч үзье.

Use case диаграм нь хэрэглэгчийн системийн талаарх ойлголтыг өгдөг. Use case диаграм нь систем, хэрэглээ, оролцогчид, харилцаа холбоо гэсэн дөрвөн бүрэлдэхүүн хэсгийг ашигладаг.

Хэрэглэгч ба системийн хамаарал



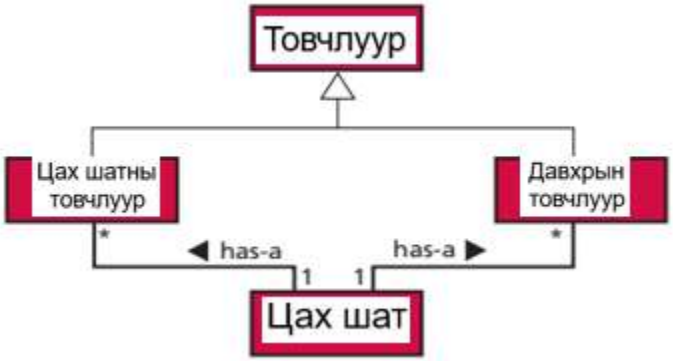
Зураг 9. use case диаграмын жишээ

Зурагт үзүүлснээр, тэгш өнцөгтөөр харуулсан хэсэг нь функцийг гүйцэтгэдэг хэсэг. Систем дэх үйлдлүүдийг бөөрөнхий тэгш өнцөгтөөр тэмдэглэсэн хэрэглээний тохиолдлуудыг харуулав.

Хэрэглэгч нь тухайн системийг ашигладаг хэн нэгэн эсвэл ямар нэгэн зүйл бөгөөд заавал хүнийг төлөөлөх албагүй.

Зураг 10.6-д өмнөх10.5-т үзүүлсэн state diagram буюу төлөв байдлын диаграммыг use case буюу хэрэглэгч болон ситсем хоорондын хамаарлын схемийг үзүүлэв. Энэ зураг дээрх систем нь цахилгаан шат юм. Цорын ганц хэрэглэгч нь лифтний хэрэглэгч юм. Хэрэглэх хоёр тохиолдол нь лифтний товчлуурыг дарах, лифт доторх шалны товчлуурыг дарах. Лифт нь давхар бүрт нэг л товчлууртай бөгөөд энэ нь лифтэнд тухайн давхарт шилжих дохио өгдөг болно.

Класс диаграм

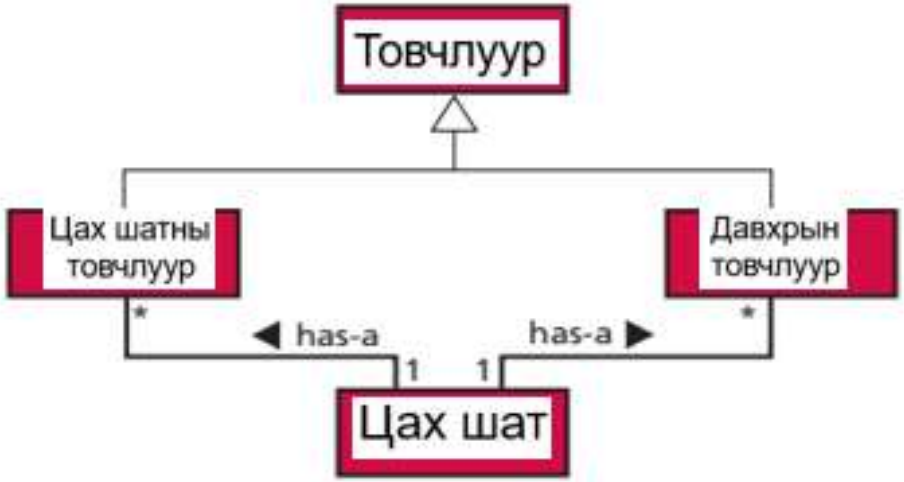


Зураг 10. Класс диаграмын жишээ

ПХ-ын анализийн дараагийн алхам бол системийг класс диаграмаар тодорхойлох юм. Жишээлбэл, бид хуучин загварын цахилгаан шатныхаа класс диаграммыг үүсгэе. Үүнд лифтийн систем нь товчлуурууд болон лифт гэсэн хоёр класстай. Тиймээс эхлээд харахад товчлуурын класс, лифтний класс гэсэн хоёр ангитай харагдаж байна. Гэхдээ цахилгаан шатны коридорын товчлуурууд болон лифт доторх шалны товчлуурууд хоёр товчлууртай. Эндээс товчлуурын класс болон товчлуурын классаас удамшсан хоёр класстай болох бөгөөд эдгээрт лифтний товчлуурын класс, шалны товчлуурын класс орно. Лифтний класс ба товчлуурын хоёр класс хооронд нэгээс олон хамааралтай.

Төлөвийн чарт

Классын бүдүүвчийг эцэслэн боловсруулсны дараа классын диаграммд класс тус бүрийн төлөвийн диаграммыг тодорхойлно. Объект хандалтат анализийн төлөвийн диаграм нь үйл ажиллагааны хандалтат анализийн төлөвийн диаграмтай ижил үүрэг гүйцэтгэдэг. Энэ нь Зураг 10.7-ийн класс диаграммын хувьд дөрвөн төлвийн дигрaмтай байх шаардлагатай гэсэн үг юм.



Зураг 11. Класс диаграмын жишээ

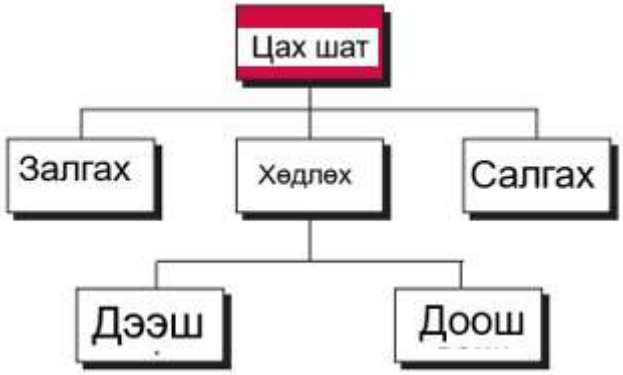
Дизайны үе шат

Дизайн үе шат (Design phase) нь шинжилгээний үе шат (Analysis phase)-д тодорхойлсон зүйлийг систем хэрхэн биелүүлэхийг тодорхойлдог. Дизайн үе шатанд системийн бүх бүрэлдэхүүн хэсгүүдийг тодорхойлдог.

Процедурын чиг баримжаатай дизайнд програм хангамжийн загварчлал болон өгөгдлийн дизайныг хийдэг. Процедурт суурилсан дизайны хувьд бүхэл системийг үйл ажиллагаа эсвэл модулиудад хуваадаг.

Бүтцийн хэсэг

Процедурт чиглэсэн дизайн дахь модулиудын хоорондын харилцааг харуулах нийтлэг хэрэгсэл бол бүтцийн диаграм юм. Жишээлбэл, 10.5-р зурагт төлөвийн ерөнхий диаграммыг харуулсан цахилгаан шатны системийн бүтцийн диаграмыг авч үзье
Зураг 10.8. Бүтцийн диаграммыг Хавсралт D-д авч үзнэ.



Зураг 12. Бүтцийн схем

МОДУЛЧЛАХ

Модульчлах гэдэг нь том төслийг ойлгож, удирдахад зориулж жижиг хэсгүүдэд хуваахыг хэлнэ. Өөрөөр хэлбэл, модульчлах гэдэг нь том ажлыг бие биетэйгээ харилцах боломжтой жижиг ажлуудад хуваах арга юм. Өмнөх хэсэгт үзсэн бүтцийн бүдүүвч нь лифтний систем дэх модулийг харуулж байна. Системийг модулиудад хуваахад хослуулах холболт ба нэгдмэл бүхэл байдал (coupling and cohesion) гэсэн хоёр үндсэн асуудлын заавал авч үзнэ.

Хөгжүүлэлт: COUPLING

Хослуулах холболт гэдэг нь хоёр модуль хоорондоо хэр нягт холбогдож байгааг харуулдаг хэмжүүр юм. Илүү нягт уялдаатай байх тусам бие даасан байдал багасна. Зорилго нь модулиудыг аль болох бие даасан болгох зорилготой бөгөөд сул чөлөөтэй холбох хэрэгтэй.
Сул чөлөөтэй холбоход хамгийн багадаа гурван шалтгаан байна:

1. Сул холбосон модулиудыг дахин ашиглах боломжтой.
2. Сул холбогдсон модулиудад алдаа гаргах магадлал бага байдаг.
3. Системийг өөрчлөх шаардлагатай үед сул холбогдсон модулиуд нь шаардлагагүй модулиудад нөлөөлөхгүйгээр зөвхөн өөрчлөх шаардлагатай модулиудыг өөрчлөх боломжийгбий болгодог.

Хөгжүүэллт

Програм хангамжийн систем дэх модулиудын холболтыг багасгах шаардлагатай.
Нэгдмэл бүхэл байдал энэ шаардлагыг биелүүлдэг.
Нэгдмэл байдал нь систем дэх модулиуд хоорондоо хэр нягт уялдаатай байгааг илэрхийлдэг хэмжүүр юм. Програм хангамжийн систем дэх модулиудын хооронд хамгийн их нийцтэй байх шаардлагатай.

Объект хандалтат шинжилгээ

Объект хандалтат дизайн нь дизайны үе шатуудад классуудын нарийвчилсан мэдээллийн боловсруулалтыг гүйцэтгэдэг.
9-р бүлэгт үзсэн класс нь хувьсагч (шинж чанар- attributes : variable) болон аргуудын (функц : functions) багцаас бүрддэг.
Зураг 10.9-д хуучин загварын лифтийг зохион бүтээхэд ашигласан манай дөрвөн класстай бүтцийг нарийвчилсан жишээг үзүүлэв.

Зураг 13. Атрибют, аргууд

Хэрэгжүүлэх үе шат, шинжилгээ

Хүрхрээ загварт дизайны үе шат дууссаны дараа хэрэгжүүлэх үе шат эхэлж болно.

Хэлний сонголт

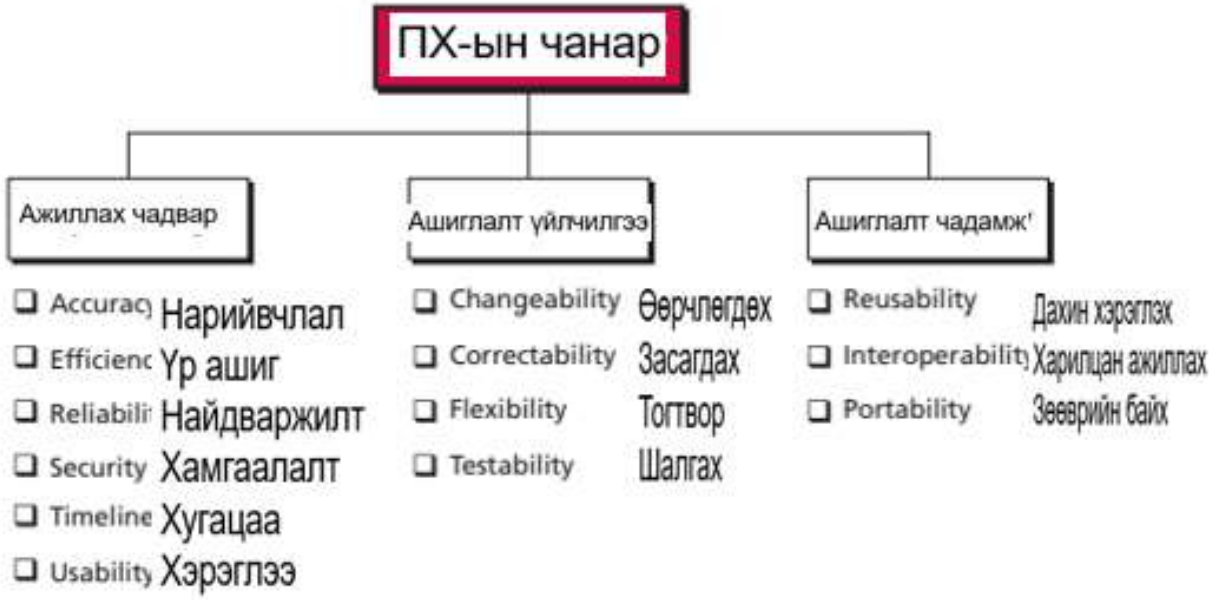
Процедурт чиглэсэн хөгжүүлэлтийн хувьд төслийн баг 9-р бүлэгт авч үзсэн процедурын хэлнүүдийн дотроос ямар нэг хэл эсвэл хэлний багцыг сонгох шаардлагатай. Хэдийгээр C++ зэрэг зарим хэлийг процедур болон объект хандалтат хэл гэж үздэг. Ихэвчлэн хэрэгжилт нь C. In гэх мэт цэвэр процедурын хэлийг ашигладаг объект хандалтат тохиолдлууд, C++ болон Java хоёулаа нийтлэг байдаг.

Програм хангамжийн чанар

Хэрэгжүүлэх үе шатанд бий болсон програм хангамжийн чанар нь хэрэглэгчийн шаардлагад нийцсэн, шаардлагыг хангасан систем юм. Гэсэн хэдий ч, хэрэв бид өндөр чанартай програм хангамжийг хөгжүүлэх үүднээс чанарын зарим шинж чанарыг тодорхойлох шаардлагатай.

ПХ-ийн чанар

Програм хангамжийн чанарыг гурван үндсэн үзүүлэлтүүдээр хуюаана: Ажиллах чадвар, Засвар үйлчилгээ, Ажиллах чадамж. Эдгээр үзүүлэлтүүдийг дараах зурагт үзүүлсэн.



Зураг 14. Чанарын үзүүлэлтүүд

Чанарын хүчин зүйлүүд

Програм хангамжийн чанарыг гурван үндсэн үзүүлэлтүүдээр хуюаана: Ажиллах чадвар, Засвар үйлчилгээ, Ажиллах чадамж. Эдгээр үзүүлэлтүүдийг дараах зурагт үзүүлсэн.

Figure 10.10 *Quality factors*

Зураг 15. Чанарын үзүүлэлтүүд

Үйл ажиллагаа

Ажиллагааны чадвар гэдэг нь системийн үндсэн ажиллагааг хэлнэ. Хэд хэдэн арга хэмжээг дурдаж болно. Дээрх зурагт үзүүлсэн шиг ажиллах чадварын хувьд нарийвчлал, үр ашиг, найдвартай байдал, аюулгүй байдал, цаг хугацаа, ашиглах боломжтой зэрэг үзүүлэлтүүдийг авч үзнэ.

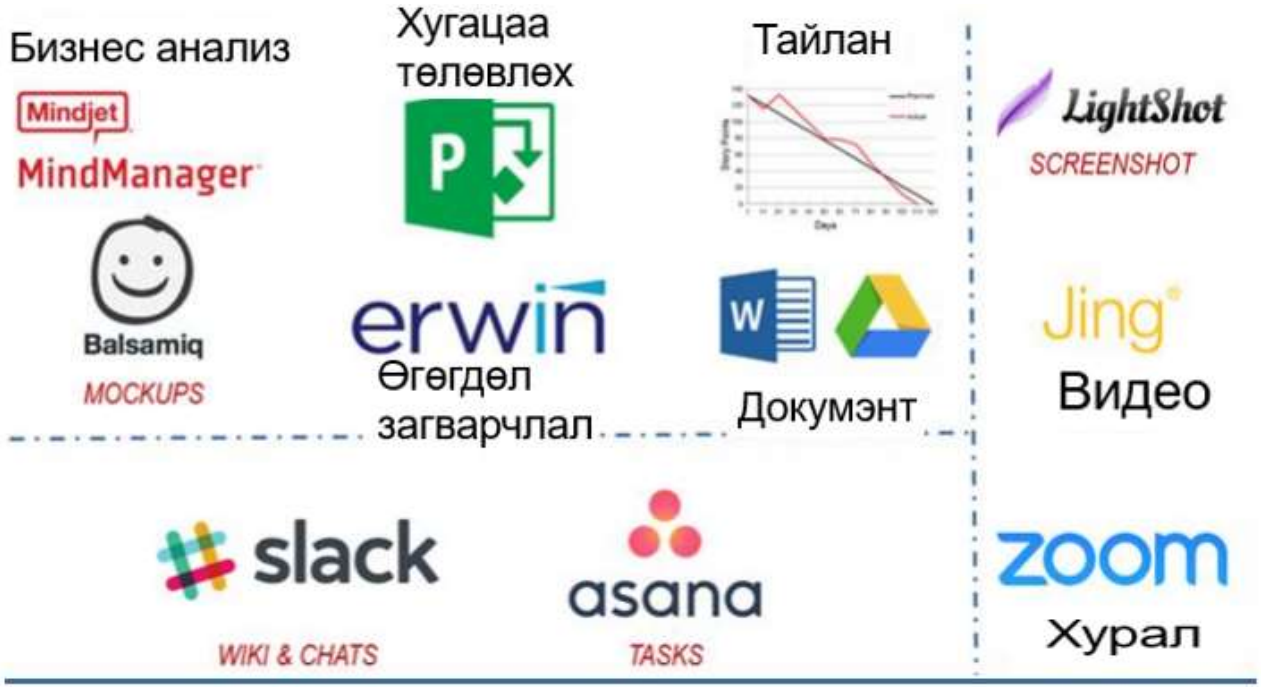
Нарийвчлалгүй систем нь огт байхгүйгээс дор юм. Тиймээс хөгжүүлсэн аливаа системийг системийн туршилтын инженер болон хэрэглэгч хоёулаа сайтар шалгаж үзэх ёстой. Нарийвчлалыг алдааны хоорондох дундаж хугацаа, кодын мянган мөрөнд ногдох алдааны тоо, өөрчлөх хүсэлтийн тоо зэрэг үзүүлэлтээр хэмжиж болно.

Үр ашиг нь субъектив нэр томъёо юм. Зарим тохиолдолд хэрэглэгч 1 секундын 95%-ийн дотор хүлээн авах бодит цагийн хариу гэх мэт гүйцэтгэлийн стандартыг зааж өгдөг. Үүнийг хэмжих нь гарцаагүй.

❑ Найдвартай байдал нь бусад хүчин зүйлсийн нийлбэр юм. Хэрэглэгчид ажлаа дуусгахын тулд системд найдаж, түүндээ итгэлтэй байвал энэ нь найдвартай байх магадлалтай. Нөгөөтэйгүүр, зарим хэмжүүрүүд нь системийн найдвартай байдалд шууд нөлөөлдөг, ялангуяа алдааны хоорондох дундаж хугацаа юм.

Хэрэгжүүлэх үе шат

❑ Систем хэр найдвартай вэ гэдэг нь зөвшөөрөлгүй хүмүүс системийн өгөгдөлд хэр хялбар нэвтэрч байгааг илэрхийлдэг. Хэдийгээр энэ нь субъектив талбар боловч системийн аюулгүй байдлыг үнэлэхэд туслах хяналтын хуудаснууд байдаг. Жишээлбэл, систем нь хэрэглэгчдийг таних нууц үгтэй бөгөөд шаарддаг уу?



Зураг 16. Чанарын үзүүлэлтүүд

Хэрэгжүүлэх үе шат

❑ Систем хэр найдвартай вэ гэдэг нь зөвшөөрөлгүй хүмүүс системийн өгөгдөлд хэр хялбар нэвтэрч байгааг илэрхийлдэг. Хэдийгээр энэ нь субъектив талбар боловч системийн аюулгүй байдлыг үнэлэхэд туслах хяналтын хуудаснууд байдаг. Жишээлбэл, систем нь хэрэглэгчдийг таних нууц үгтэй бөгөөд шаарддаг уу?

❑ Програм хангамжийн инженерчлэлд цаг үеэ олсон байх нь хэд хэдэн өөр зүйлийг илэрхийлж болно. Системийг хийдэгбүтээгдэхүүнээ цаг тухайд нь хүргэх үү? Онлайн системийн хувьд хариу өгөх хугацаа нь хэрэглэгчдийн шаардлагад нийцэж байна уу?

❑ Ашиглах боломж бол маш субъектив талбар юм. Ашиглалтын хамгийн сайн хэмжүүр бол хэрэглэгчдийг ажиглаж, системийг хэрхэн ашиглаж байгааг харах явдал юм. Хэрэглэгчийн ярилцлага нь системийн ашиглалттай холбоотой асуудлуудыг ихэвчлэн илрүүлдэг.

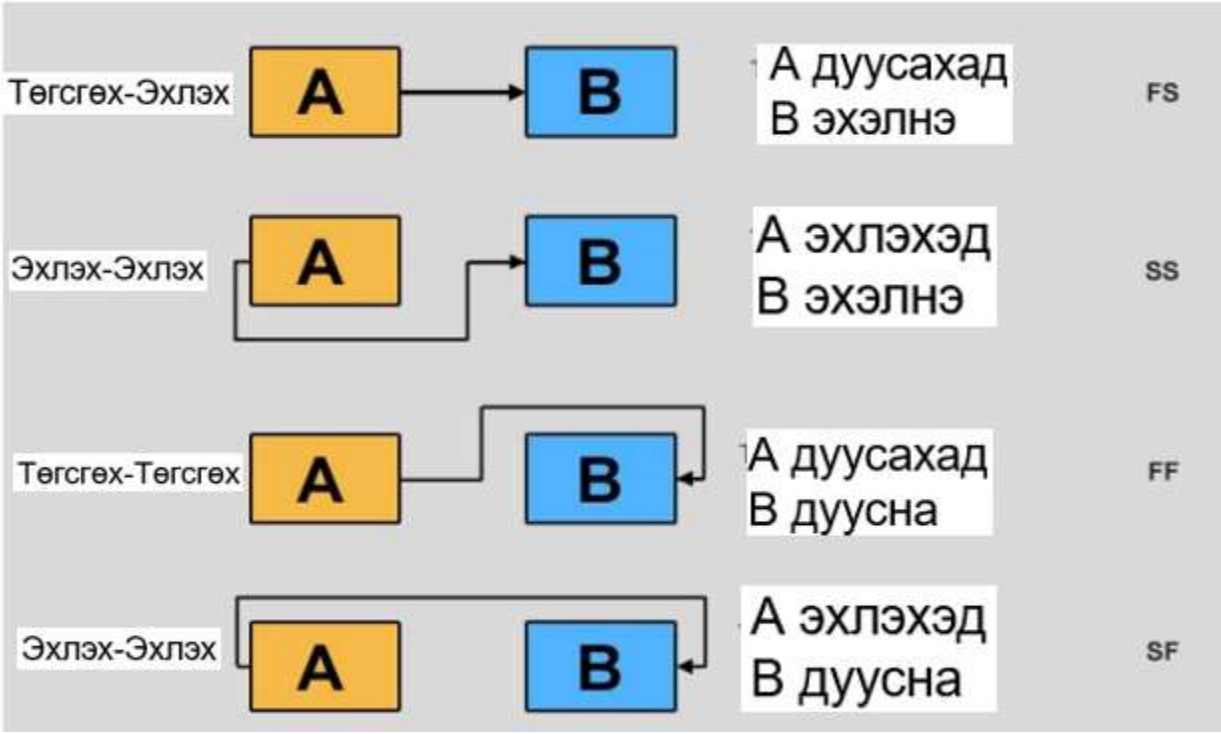
Хэрэгжүүлэх үе шат

Тогтвортой байдал

Засвар үйлчилгээ гэдэг нь системийг шинэчилж, зөв ажиллуулахад хялбар байхыг хэлнэ. Олон систем нь муу хэрэгжсэнээсээ биш, харин гадны хүчин зүйлийн өөрчлөлтөөс болж байнгын өөрчлөлтийг шаарддаг. Жишээлбэл, засгийн газрын хууль тогтоомжийн өөрчлөлтөд нийцүүлэн компанийн цалингийн системийг байнга өөрчлөх шаардлагатай болдог.

Өөрчлөгдөх чадвар нь субъектив хүчин зүйл юм. Туршлагатай төслийн удирдагчид хүссэн өөрчлөлтийг хэрэгжүүлэхэд хэр хугацаа шаардагдахыг тооцоолж чаддаг. Хэрэв хэтэрхий урт бол энэ нь системийг өөрчлөхөд хэцүү байгааг илтгэнэ. Энэ нь ялангуяа хуучин системүүдийн хувьд үнэн юм. Өнөөдөр энэ салбарт програмын нарийн төвөгтэй байдал, бүтцийг тооцоолох програм хангамжийн хэмжилтийн хэрэгслүүд байдаг.

Төслийн хамаарал ба цаг хугацаа



Зураг 17. Програмын хөгжүүлэлтийн үеийн хамаарал

Залруулж болохуйц нэг хэмжүүр бол програм бүтэлгүйтсэний дараа сэргээхэд шаардагдах дундаж хугацаа юм. Хэдийгээр энэ нь реактив тодорхойлолт боловч хөтөлбөр бүтэлгүйтсэн үед түүнийг засахад хэр хугацаа шаардагдахыг урьдчилан таамаглах зүйл одоогоор алга байна.

Хэрэгжүүлэх үе шат

❑ Хэрэглэгчид системд өөрчлөлт оруулах хүсэлтийг байнга тавьдаг. Уян хатан байдал нь эдгээр өөрчлөлтийг хийхэд хэр хялбар болохыг хэмжихийг оролддог чанарын шинж чанар юм. Хэрэв өөрчлөлт оруулахын тулд програмыг бүрэн дахин бичих шаардлагатай бол энэ нь уян хатан биш юм.

Зураг 18. Чанарын үзүүлэлтүүд

Хэрэгжүүлэх үе шат

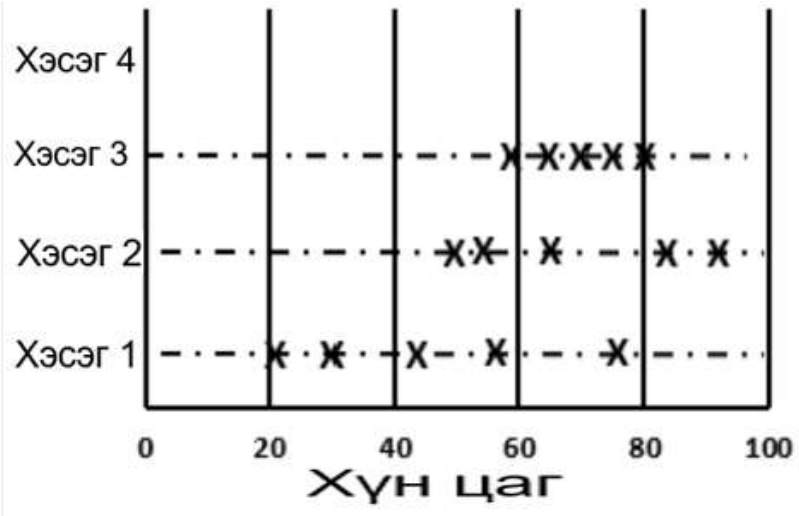
❑ Хэрэглэгчид системд өөрчлөлт оруулах хүсэлтийг байнга тавьдаг. Уян хатан байдал нь эдгээр өөрчлөлтийг хийхэд хэр хялбар болохыг хэмжихийг оролддог чанарын шинж чанар юм. Хэрэв өөрчлөлт оруулахын тулд програмыг бүрэн дахин бичих шаардлагатай бол энэ нь уян хатан биш юм.

❑ Туршилтыг маш субьектив талбар гэж бид бодож болох ч туршилтын инженерүүд хөтөлбөрийн туршилтын чадварыг үнэлэх хүчин зүйлсийн хяналтын хуудастай байдаг.Шилжүүлэх чадварДамжуулах чадвар гэдэг нь өгөгдөл болон/эсвэл системийг нэг платформоос нөгөө платформ руу шилжүүлэх, кодыг дахин ашиглах чадварыг хэлнэ. Ихэнх тохиолдолд энэ нь чухал хүчин зүйл биш юм. Нөгөөтэйгүүр, хэрэв бид ерөнхий програм хангамж бичиж байгаа бол энэ нь шүүмжлэлтэй байж болно.

❑ Хэрэв модулиуд нь бусад системд дахин ашиглагдахаар бичигдсэн бол дахин ашиглах чадвар өндөр байна. Сайн програмистууд ижил төстэй асуудлуудыг шийдвэрлэхэд дахин ашиглах боломжтой функцүүдийн санг бий болгодог.

Хэрэгжүүлэх үе шат

- Харилцан ажиллах чадвар нь бусад системд өгөгдөл дамжуулах чадвар юм. Өнөөгийн өндөр нэгдсэн системд энэ нь зүйтэй шинж чанар юм. Үнэн хэрэгтээ энэ нь маш чухал болсон үйлдлийн системүүд одоо систем хооронд өгөгдөл шилжүүлэх чадварыг дэмждэгтекст процессор болон хүснэгтийн хооронд.
- Зөөврийн чадвар гэдэг нь програм хангамжийг нэг техник хангамжийн платформоос нөгөөд шилжүүлэх чадвар юм.

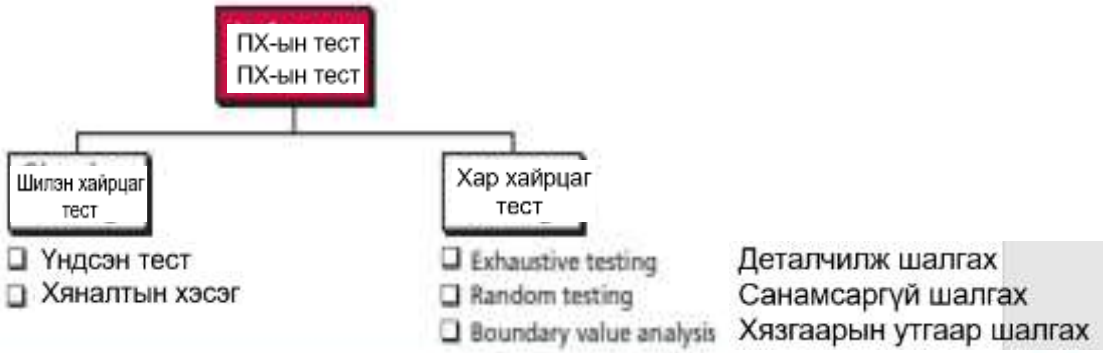


Зураг 19. Баг сонгох, уулзалт, хувь хүн, үнэлгээ, ажлууд, үр дүн

Тестлэлт

Туршилтын үе шатны зорилго нь алдааг олох явдал бөгөөд энэ нь сайн туршилтын стратеги нь ихэнх алдааг олж илрүүлдэг гэсэн үг юм. Шинжилгээний хоёр төрөл байдаг: шилэн хайрцаг ба хар хайрцаг

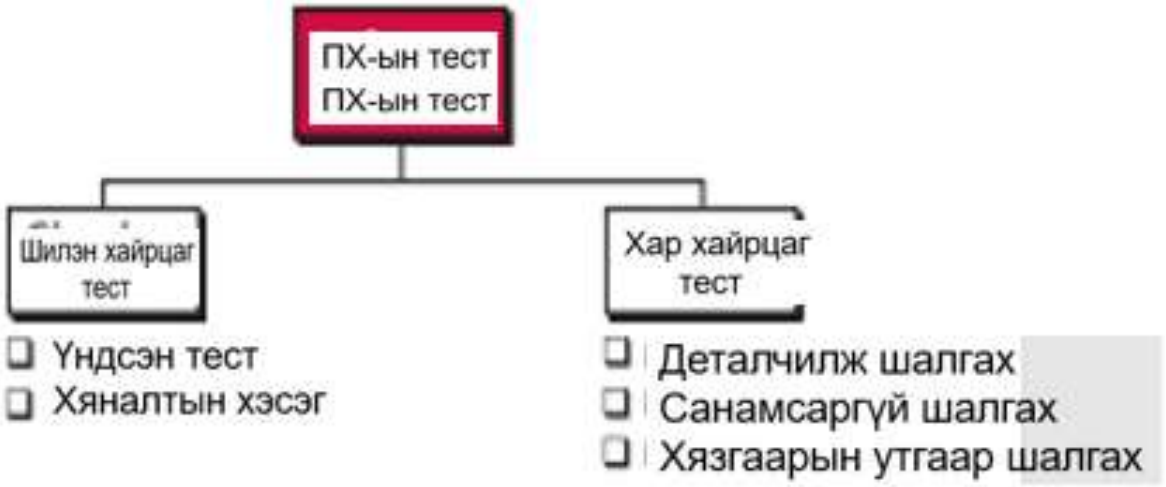
Figure 10.11 *Software testing*



Зураг 20. Тестлэлт

Тестлэлтийн үе шат

Туршилтын үе шатны зорилго нь алдааг олох явдал бөгөөд энэ нь сайн туршилтын стратеги нь ихэнх алдааг олж илрүүлдэг гэсэн үг юм. Шинжилгээний хоёр төрөл байдаг: шилэн хайрцаг ба хар хайрцаг



Зураг 21. ПХ-ын тестлэлт

Тестлэлтийн үе шат

10.5.1 Glass-box testing (Шилэн хайрцагны туршилт)

1950 оноос хойш бүтээгдсэн компьютерууд вон Нейманы загварыг дагаж мөрддөг. Тэд илүү хурдан, жижиг, хямд болсон боловч зарчим нь бараг ижил юм. Түүхчид энэ үеийг үеүдэд хуваадаг бөгөөд үе бүр техник хангамж эсвэл программ хангамжийн томоохон өөрчлөлтийг гэрчилдэг (гэхдээ загварт биш).

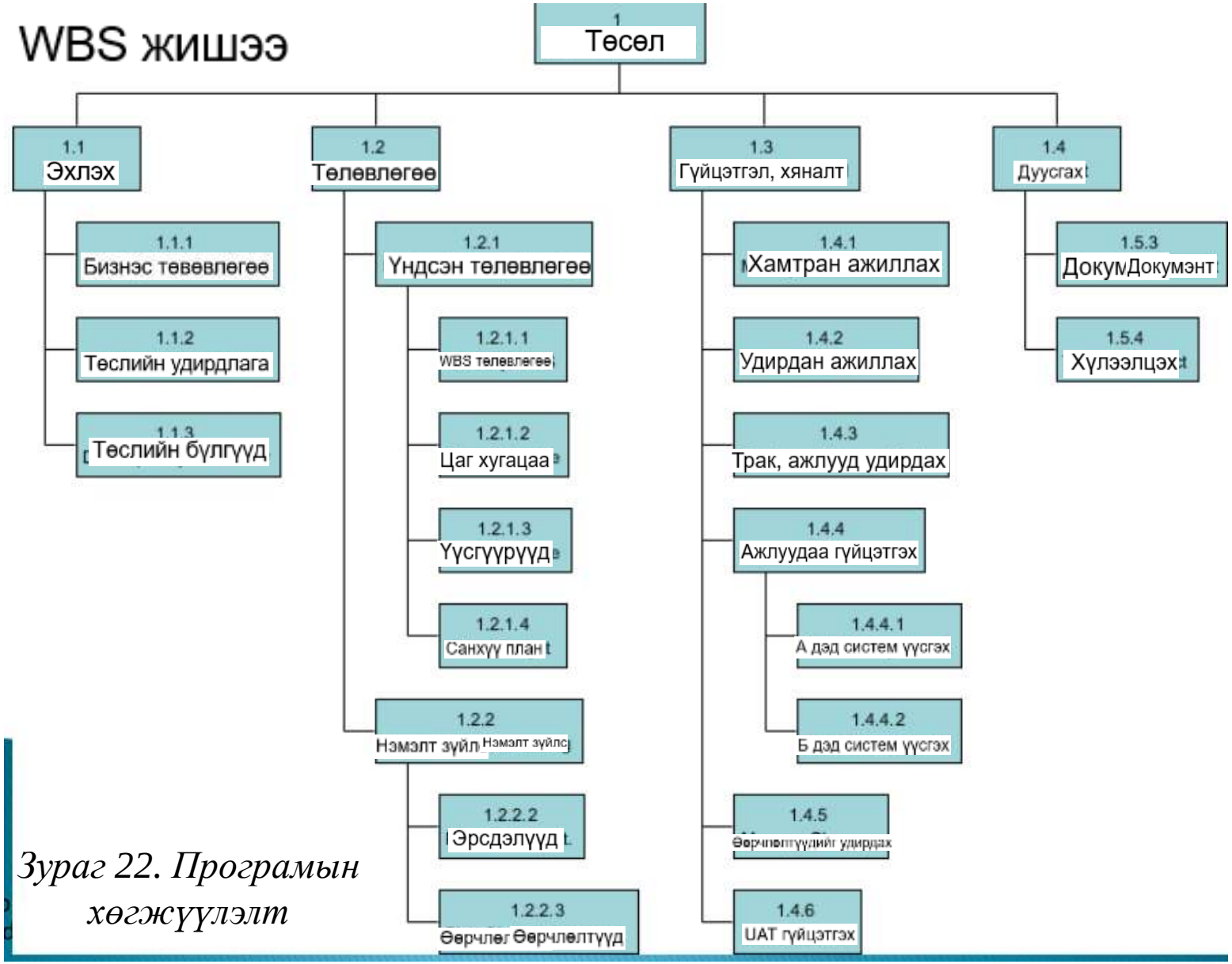
Шилэн хайрцагны туршилт (эсвэл цагаан хайрцагны туршилт) нь програм хангамжийн дотоод бүтцийг мэдэхэд суурилдаг. Туршилтын зорилго нь програм хангамжийн бүх бүрэлдэхүүн хэсгүүд хийхээр төлөвлөж буй зүйлээ хийж байгаа эсэхийг шалгах явдал юм. Шилэн хайрцагны туршилт нь шалгагч програм хангамжийн талаар бүгдийг мэддэг гэж үздэг. Энэ тохиолдолд програм хангамж нь хайрцагны доторх бүх зүйл харагдахуйц шилэн хайрцагтай адил юм. Шилэн хайрцагны туршилтыг програм хангамжийн инженер эсвэл тусгай баг хийдэг. Дор хаяж дараах дөрвөн шалгуурыг хангасан эсэхийг баталгаажуулахын тулд програм хангамжийн бүтцийг ашигладаг шилэн хайрцагны туршилт хийх шаардлагатай..

Тестлэлтийн үе шат

- Модуль бүрийн бие даасан бүх замыг дор хаяж нэг удаа шалгана.
- Бүх шийдвэрийн бүтцийг (хоёр болон олон талт) салбар бүр дээр туршина.
- Гогцооны бүтэц бүрийг туршина.
- Бүх өгөгдлийн бүтцийг туршсан.Өнгөрсөн хугацаанд хэд хэдэн туршилтын аргачлалыг боловсруулсан.

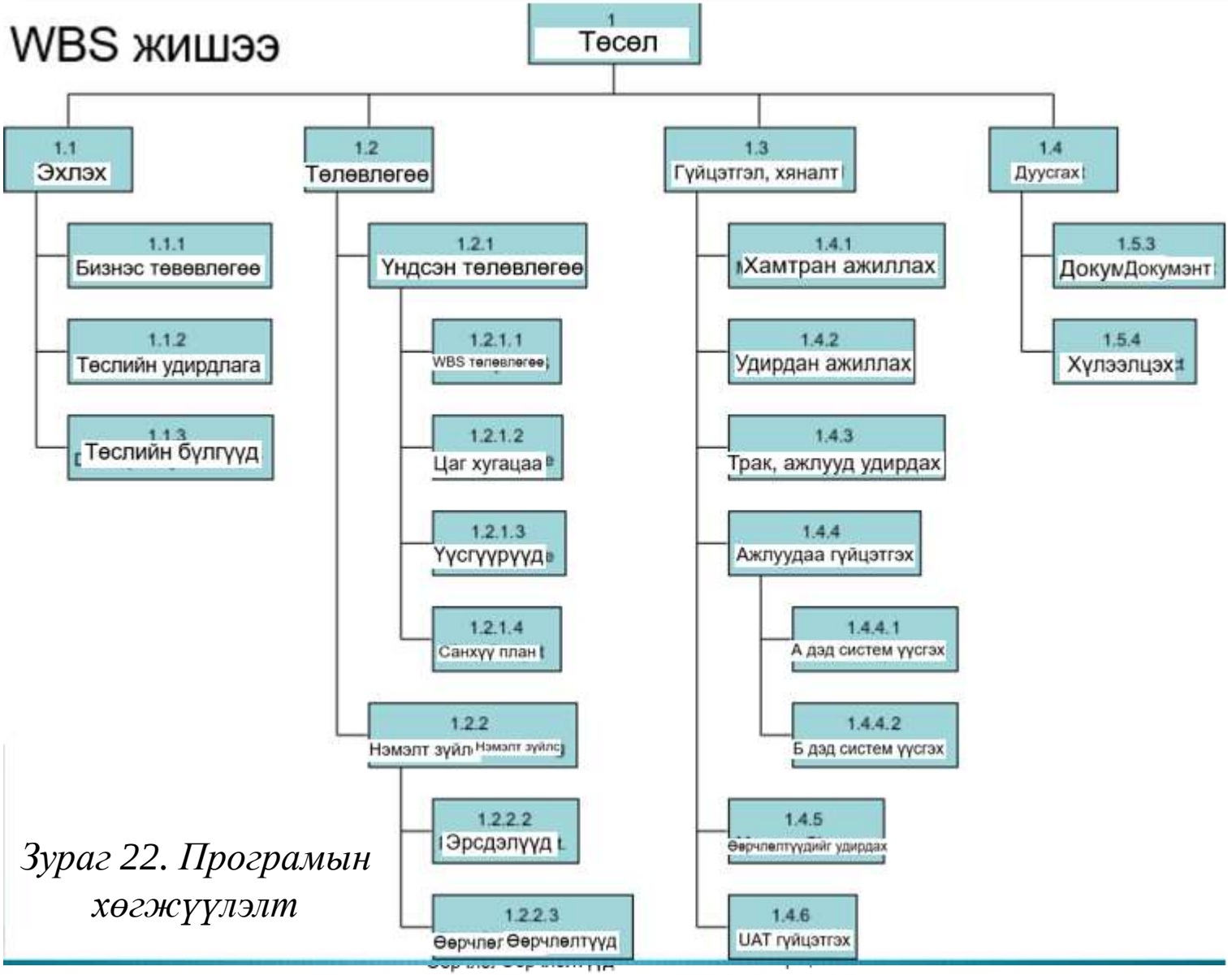
Бид хоёрын талаар товч ярилцанатэдгээрт: үндсэн замын туршилт ба хяналтын бүтцийн туршилт.

Жишээ



Зураг 22. Програмын хөгжүүлэлт

Жишээ



Зураг 22. Програмын хөгжүүлэлт

Тестлэлтийн үе шат

Замын үндсэн туршилт

Замын үндсэн туршилтыг Том МакКэйб санал болгосон. Энэ арга нь программ хангамж дахь мэдэгдэл бүрийг дор хаяж нэг удаа гүйцэтгэх тестийн багцыг үүсгэдэг.

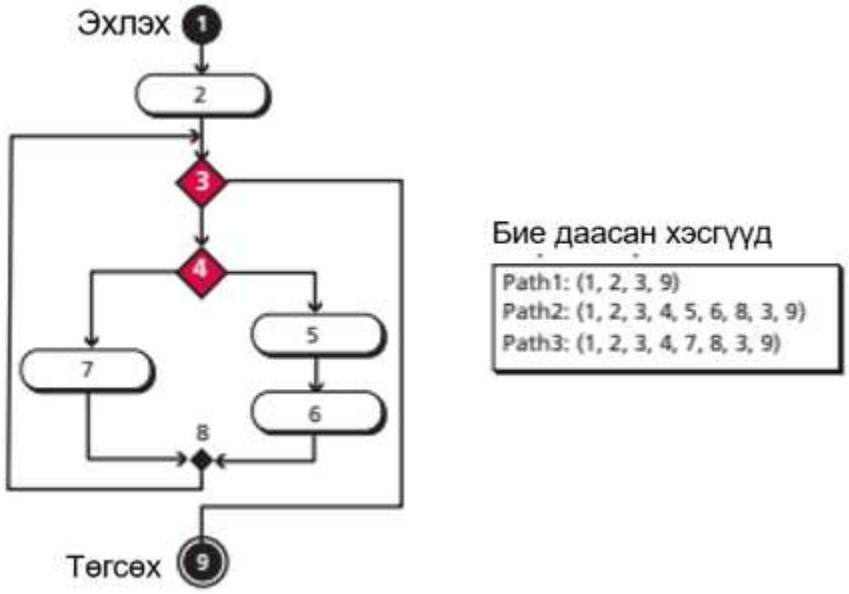
Суурийн замын тест гэдэг нь програм хангамжийн мэдэгдэл бүрийг дор хаяж нэг удаа гүйцэтгэдэг арга юм.

Суурийн замын туршилт нь графын онол (12-р бүлгийг үзнэ үү) болон мөчлөгийн нарийн төвөгтэй байдлыг ашиглан мэдэгдэл бүрийг дор хаяж нэг удаа гүйцэтгэх баталгааг хангахын тулд дагаж мөрдөх ёстой бие даасан замыг олох болно.

Тестлэлтйн үе шат

Жишээ 10.1

Програмын нэг хэсэг дэх үндсэн замыг шалгах, бие даасан замыг олох санааг өгөхийн тулд систем нь зөвхөн нэг програмаас бүрдэх ба уг программ нь зөвхөн нэг давталт бөгөөд Зураг 10.12-т үзүүлсэн UML диаграммтай гэж үзье.



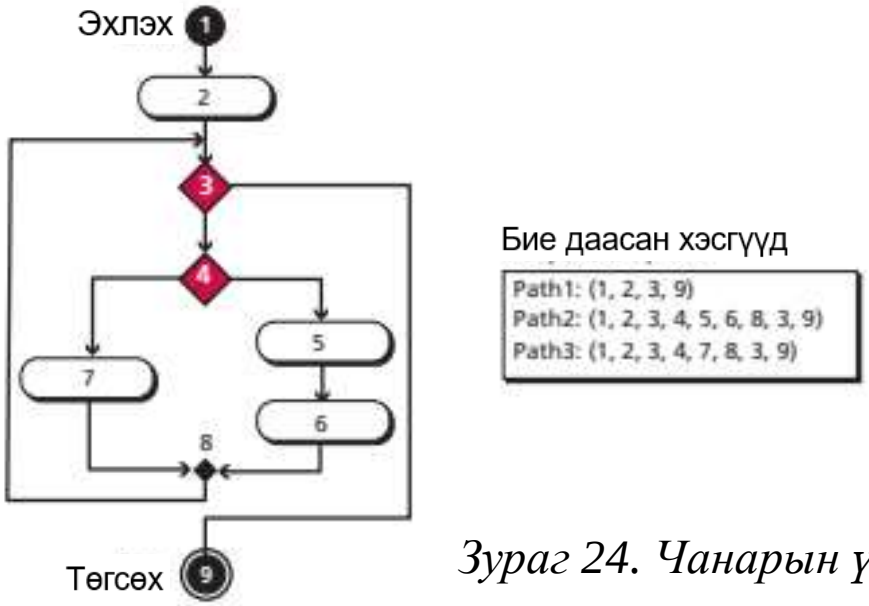
Зураг 23. Үндсэн тест

Тестлэлтйн үе шат

Жишээ 10.1

Програмын нэг хэсэг дэх үндсэн замыг шалгах, бие даасан замыг олох санааг өгөхийн тулд систем нь зөвхөн нэг програмаас бүрдэх ба уг программ нь зөвхөн нэг давталт бөгөөд Зураг 10.12-т үзүүлсэн UML диаграммтай гэж үзье.

Figure 10.12 *An example of basis path testing*



Зураг 24. Чанарын үзүүлэлтүүд

Тестлэлтйн үе шат

Энэхүү энгийн программд бид гурван бие даасан замтай. Эхний зам нь гогцоог тойрч гарах тохиолдол юм. Хоёр дахь зам нь давталтыг нэг удаа гүйцэтгэнэ шийдвэрийн бүтцийн зөв салбар. Гурав дахь зам нь давталтыг нэг удаа гүйцэтгэх боловч шийдвэрийн бүтцийн зүүн салбараар дамжих явдал юм. Хэрэв илүү олон давталт байгаа бол үүсгэсэн замууд нь эдгээр гурван замаас хамааралгүй: хэрэв бид UML-ийг урсгалын график болгон өөрчилвөл үүнийг батлах боломжтой, гэхдээ бид нотлох баримтыг програм хангамжийн инженерчлэлийн номонд үлдээх ёстой. Энэхүү санаа нь үндсэн замын багц дахь бүх гурван замыг хамарсан туршилтын тохиолдлуудыг зохион бүтээх явдал юм бүх мэдэгдлийг дор хаяж нэг удаа гүйцэтгэнэ.

Тестлэлтйн үе шат

Control structure testing

Хяналтын бүтцийн туршилт нь үндсэн замын туршилтаас илүү өргөн хүрээтэй бөгөөд үүнд багтдаг. Энэ арга нь доор товч тайлбарласан янз бүрийн ангиллын тестийг ашигладаг.

Condition testing

Нөхцөл байдлын туршилт нь модуль дахь аливаа нөхцөл байдлын илэрхийлэлд хамаарна. Энгийн нөхцөл бол харилцааны илэрхийлэл бол нийлмэл нөхцөл нь энгийн нөхцлүүдийн нэгдэл юмба логик операторууд Нөхцөл байдлын туршилт нь бүх нөхцөлийг зөв тохируулсан эсэхийг шалгах зорилготой юм.

Data flow testing

Өгөгдлийн урсгалын тест нь модулаар дамжих өгөгдлийн урсгал дээр суурилдаг. Энэ төрлийн тест нь даалгаврын мэдэгдлийн зүүн талд ашиглагдаж байгаа хувьсагчдын утгыг шалгахтай холбоотой тест тохиолдлуудыг сонгодог.

Тестлэлтйн үе шат

Loop testing

Давталтын тест нь гогцооны хүчинтэй эсэхийг шалгахын тулд туршилтын тохиолдлуудыг ашигладаг. Бүх төрлийн гогцоог (while, do, for) сайтар шалгадаг.

10.5.2 Black-box testing

Хар хайрцагны тест гэдэг нь программ хангамжийг дотор нь юу байгааг, хэрхэн ажилладагийг нь мэдэхгүйгээр турших гэсэн ойлголтоос нэрээ авсан. Өөрөөр хэлбэл, программ хангамж нь шалгагч хардаггүй хар хайрцагтай адил юм. Хар хайрцагны тест нь програм хангамжийн оролт, гаралт гэх мэт юу хийх ёстойг харгалзан програм хангамжийн ажиллагааг шалгадаг. Хар хайрцагны туршилтанд хэд хэдэн аргыг ашигладаг бөгөөд үүнийг доор авч үзэх болно.

Тестлэлтйн үе шат

Exhaustive testing

Хар хайрцгийг шалгах хамгийн сайн арга бол програм хангамжийг оролтын домайн дахь боломжит бүх утгыг шалгах явдал юм. Гэсэн хэдий ч нарийн төвөгтэй програм хангамжийн оролтын домэйн нь маш том тул үүнийг хийх нь ихэвчлэн боломжгүй байдаг.

Random testing

Санамсаргүй туршилтын хувьд оролтын домайн дахь утгуудын дэд багцыг туршихаар сонгоно. Домэйн оролт дээр утгууд нь тархсан байхаар дэд олонлогийг сонгох нь маш чухал юм. Санамсаргүй тоо үүсгэгчийг ашиглах нь маш их тустай байж болно

Энэ тохиолдолд.

Boundary-value testing Хилийн утгуудтай тулгарах үед алдаа ихэвчлэн гардаг. Жишээлбэл, хэрэв модуль нь түүний оролтын аль нэг нь 100-аас их эсвэл тэнцүү байх ёстой гэж тодорхойлсон бол модулийг 100-ийн хилийн утгыг шалгах нь маш чухал юм. Хэрэв модуль энэ хязгаарт амжилтгүй болбол утга байгаа бол модулийн кодын зарим нөхцөл тухайлбал x > 100-г x<100 гэж бичнэ.

Экстим програмчлал арга зүй

Зураг 25. Програмын хөгжүүлэлт

Бичиг баримт

Програм хангамжийг зөв ашиглаж, үр ашигтай байлгахын тулд баримт бичиг шаардлагатай.
Ихэвчлэн програм хангамжид зориулж гурван тусдаа баримт бичгийг бэлтгэдэг:
хэрэглэгчийн баримт бичиг,
системийн баримт бичиг,
техникийн баримт бичиг.
Бүх баримт бичиг нь тасралтгүй үйл явц.
Хэрэв програм хангамж гарсны дараа асуудал гарвал тэдгээрийг мөн баримтжуулсан байх ёстой. Хэрэв програм хангамж өөрчлөгдсөн бол бүх өөрчлөлтүүд болон тэдгээрийн анхны багцтай харьцах харьцааг баримтжуулсан байх ёстой.
Баримт бичгийг багцлах үед л зогсдог хуучирдаг.

Бичиг баримт

Хэрэглэгчийн баримт бичиг

Програм хангамжийн системийг зөв ажиллуулахын тулд хэрэглэгчдэд программ хангамжийг хэрхэн ашиглахыг алхам алхмаар харуулсан баримт бичиг хэрэгтэй бөгөөд үүнийг хэрэглэгчийн гарын авлага гэж нэрлэдэг. Хэрэглэгчийн гарын авлага нь ихэвчлэн програм хангамжийн онцлог бүрээр дамжуулан хэрэглэгчийг чиглүүлэх зааварчилгааны хэсгийг агуулдаг. Сайн хэрэглэгчийн гарын авлага нь маркетингийн маш хүчирхэг хэрэгсэл байж болно: маркетингийн хэрэглэгчийн баримт бичгийн ач холбогдлыг онцлон тэмдэглэж болохгүй. Хэрэглэгчийн гарын авлагыг шинэхэн болон туршлагатай хэрэглэгчдэд зориулж бичсэн байх ёстой бөгөөд хэрэглэгчийн сайн баримт бичиг бүхий програм хангамжийн систем нь борлуулалтыг нэмэгдүүлэх нь гарцаагүй.

Бичиг баримт

Системийн баримт бичиг нь програм хангамжийг тодорхойлдог.
Анхны хөгжүүлэгчээс бусад хүмүүс програм хангамжийг засварлаж, өөрчлөх боломжтой байхаар бичсэн байх ёстой.
Системийн баримт бичиг нь системийн хөгжлийн бүх дөрвөн үе шатд хуваана.
Шинжилгээний үе шатанд цуглуулсан мэдээллийг сайтар баримтжуулсан байх ёстой. Үүнээс гадна шинжээчид мэдээллийн эх сурвалжийг тодорхойлох ёстой. Энэ үе шатанд сонгогдсон шаардлага, аргачлалууд нь тэдгээрийн үндэслэлийг тодорхой тусгасан байх ёстой.
Загварын үе шатанд эцсийн хуулбарт ашигласан багаж хэрэгслийг баримтжуулсан байх ёстой. Жишээлбэл, диаграмд хэд хэдэн өөрчлөлт орсон бол диаграмын эцсийн хуулбарыг бүрэн тайлбарын хамт баримтжуулсан байх ёстой.
Хэрэгжүүлэх үе шатанд кодын модул бүрийг баримтжуулсан байх ёстой. Нэмж дурдахад, код нь тайлбар болон тайлбарын толгойг ашиглан аль болох өөрөө баримтжуулсан байх ёстой.
Туршилтын үе шатыг сайтар баримтжуулах ёстой. Эцсийн бүтээгдэхүүнд ашигласан туршилтын төрөл бүрийг үр дүнгийн хамт дурдах ёстой. Бүр таагүй үр дүн, түүнийг үүсгэсэн өгөгдлийг баримтжуулсан байх ёстой.

Бичиг баримт

Техникийн баримт бичигт програм хангамжийн системийг суурилуулах, засвар үйлчилгээ хийх талаар тайлбарласан болно. Суулгах баримт бичиг нь сервер, үйлчлүүлэгч гэх мэт компьютер бүрт програм хангамжийг хэрхэн суулгахыг тодорхойлдог. Үйлчилгээний баримт бичиг нь системийг хэрхэн хадгалах, шаардлагатай бол шинэчлэхийг тодорхойлдог.

Төслийн ажил

❑ Програм хангамжийн амьдралын мөчлөг нь програм хангамжийн инженерчлэлийн үндсэн ойлголт юм. Програм хангамж нь бусад олон бүтээгдэхүүний нэгэн адил давтагдах үе шатуудыг дамждаг.

❑ Програм хангамжийн амьдралын мөчлөгийг боловсруулах үйл явц нь шинжилгээ, дизайн, хэрэгжилт, туршилт гэсэн дөрвөн үе шатыг агуулдаг. Эдгээр үе шаттай холбоотой хэд хэдэн загварыг ашигласан.Бид хамгийн түгээмэл хоёрыг авч үзсэн: хүрхрээ загвар ба нэмэгдэл загвар.

❑ Хөгжлийн үйл явц нь шинжилгээний үе шатаас эхэлдэг. Шинжээч нь програм хангамжийг хэрхэн яаж хийхийг зааж өгөхгүйгээр юу хийхийг харуулсан тодорхойлолтын баримт бичгийг бэлтгэдэг. Шинжилгээний үе шатыг процедурт чиглэсэн шинжилгээ, объект хандалтат шинжилгээ гэсэн хоёр аргаар хийж болно.

❑ Дизайн үе шат нь шинжилгээний үе шатанд тодорхойлсон зүйлийг систем хэрхэн биелүүлэхийг тодорхойлдог. Процедурын чиг баримжаатай дизайны хувьд бүх төслийг процедур эсвэл модулиудын багцад хуваадаг. Объект хандалтат дизайны хувьд дизайны үе шат нь ангиудын дэлгэрэнгүй мэдээллийг боловсруулах замаар үргэлжилдэг.

END-CHAPTER MATERIALS

❑ Модульчлага гэдэг нь том төслийг хялбархан ойлгож, зохицуулах боломжтой жижиг хэсгүүдэд хуваахыг хэлнэ. Системийг модулиудад хуваахад хоёр асуудал чухал байдаг: холболт ба нэгдмэл байдал. Холболт гэдэг нь хоёр модуль хоорондоо хэр нягт холбогдож байгааг харуулдаг хэмжүүр юм. Програм хангамжийн систем дэх модулиудын холболтыг багасгах шаардлагатай.Нэгдмэл байдал нь систем дэх модулиуд хоорондоо хэр нягт уялдаатай байгааг илэрхийлдэг хэмжүүр юм. Програм хангамжийн систем дэх модулиудын хоорондын уялдаа холбоог дээд зэргээр нэмэгдүүлэх шаардлагатай.

❑ Хэрэгжүүлэх үе шатанд программистууд модулиудын кодыг процедур хандалтат загварт бичих эсвэл объект хандалтат дизайн дахь ангиудыг хэрэгжүүлэх програмын нэгжүүдийг бичдэг.

❑ Програм хангамжийн чанар чухал. Програм хангамжийн чанарыг гурван том хэмжүүрт хувааж болно: ажиллах чадвар, засвар үйлчилгээ, дамжуулах чадвар.

Төслийнм ажлын талаар

❑ Туршилтын үе шатны зорилго нь алдааг олох явдал юм. Шинжилгээ нь шилэн хайрцаг, хар хайрцаг гэсэн хоёр төрөлтэй. Шилэн хайрцагны туршилт (эсвэл цагаан хайрцагны туршилт) нь үүнийг мэдэхэд суурилдагпрограм хангамжийн дотоод бүтэц. Шилэн хайрцагны туршилт нь шалгагч бүх зүйлийг мэддэг гэж үздэг. Хар хайрцагны тест гэдэг нь програм хангамжийг дотор нь юу байгааг мэдэхгүй, хэрхэн ажилладагийг нь мэдэхгүйгээр туршихыг хэлнэ.

АШИГЛАСАН МАТЕРИАЛ

Foundations of Computer Science, Behrouz A. Forouzan, Fourth Edition, Cengage Learning EMEA, 2018

Tuyatsetseg Badarch, "Data communications and computer networking" , third edition, 2014.

Pressman, R. Software Engineering: A Practitioner’s Approach, Нью-Йорк: McGraw-Hill, 2005 он

Schach, S. Объект хандалтат ба сонгодог програм хангамжийн инженерчлэл, Нью-Йорк: МакГроу-Хилл,
2007 он

АНХААРАЛ ХАНДУУЛСАНД БАЯРЛАЛАА