

Архитектурын харагдацууд

Програм хангамжийн архитектур (Software Architecture)

2012

С. Бадрал

Агуулга

- Нэр томьёо
- Харагдац гэж юу вэ?
- Яагаад?
- Мэргэжлийн харагдац
- Статик харагдац
- Статик харагдацыг бүтэцлэхэд юу туслах вэ?
- Ажиллах цагийн харагдац
- Суурилуулалтын харагдац
- Дүгнэлт

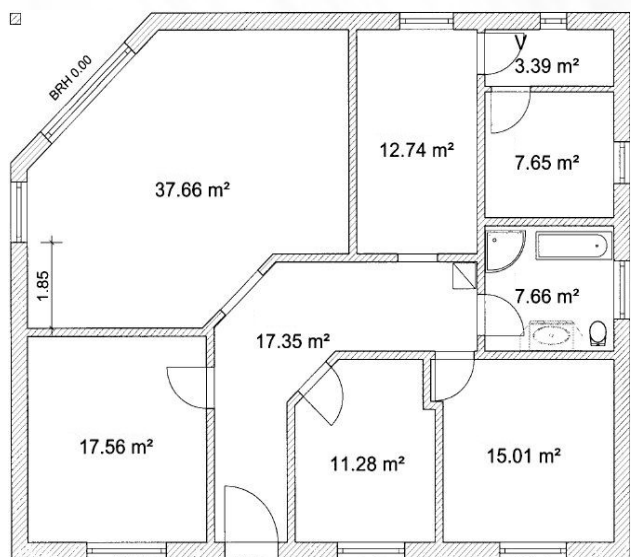
Нэр томьёо

- View = Харагдац
- Architecture style = Архитектурын хэлбэр
- Design pattern = Зохиомжийн загвар
- Static view = Статик харагдац
- Runtime view = Ажиллах үеийн харагдац
- Deployment view = Суурилуулалтын харагдац
- Technical view = Мэргэжлийн харагдац
- Responsibility = Хариуцлага
- Cyclomatic complexity = Цикломатик нийлмэл байдал

Харагдац гэж юу вэ?

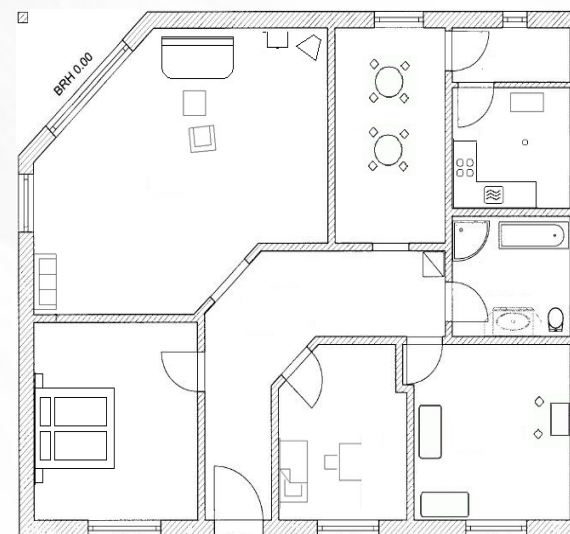


Нүүрний 2D харагдац



Дэвсгэр зураг

Барилгын Архитектурын Харагдац



Өрөөний зохион байгуулалт

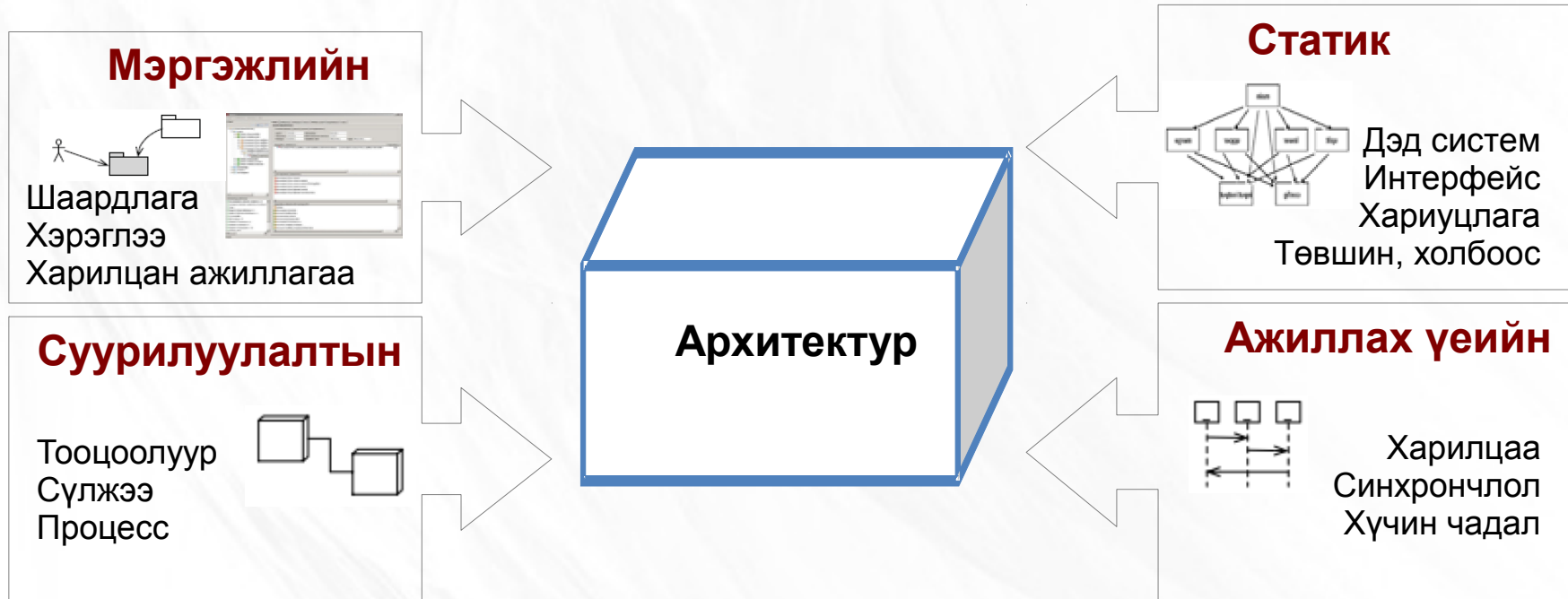
Яагаад харагдац гэж?

Програм хангамжийн архитектурын хувьд

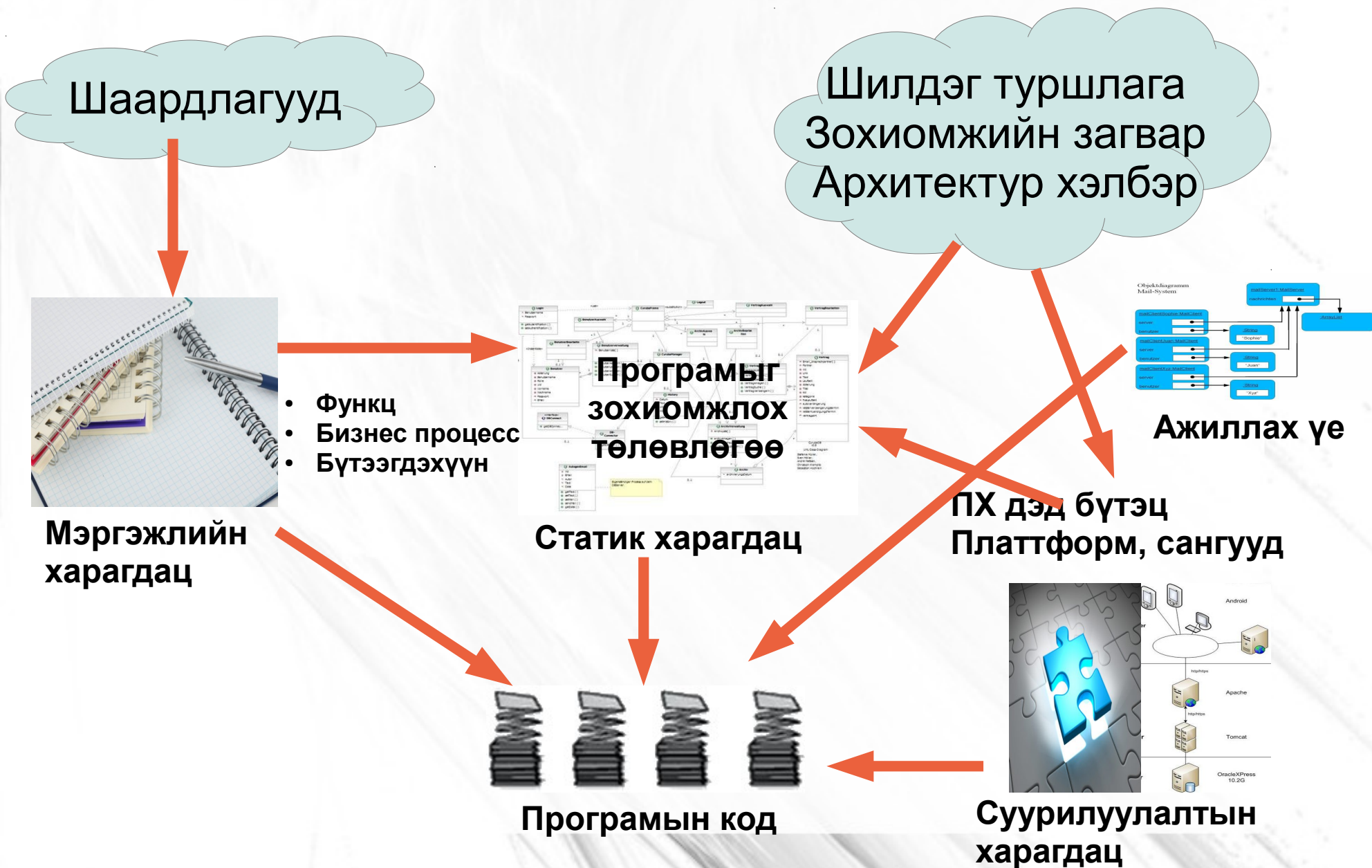
- ПХ олон талын шаардлагатай, төвөгтэй бүтэцтэй, хэрэглээнд нэвтрүүлэхэд амаргүй
- Янз бүрийн олон оролцогчидтой (үйлчлүүлэгч, эцсийн хэрэглэгч, зөвлөх, хөгжүүлэгч, захирагч,..)
- Хэрэглэх болон хөгжүүлэхэд тунгалаг байх ёстой
- Харагдац нь нэгж байдалд төвлөрөх боломж олгоно
- Харагдац нь ойлгомжтой байдлыг өсгөнө
- Харагдац нь төсөлд оролцогчдын мэдээллийн янз бүрийн хэрэгцээг хангана

ПХ Харагдацын загвар

- 4 хэлбэр



Харагдац төсөлд



Харагдацын зохиомж

Харагдацын дараалал

- Статик, хэрэв төстэй систем хэдийнээ хөгжүүлэгдсэн бол эхлээд түүнийг харах
- Ажиллах үеийн, хэрэв элементүүдийн харилцан ажиллагааг тодруулах бол
- Суурилуулалтын, хэрэв техникийн дэд бүтцийн улмаас янз бүрийн нөхцөл байдал үүсвэл
- Мэргэжлийн, бүх тохиолдолд

Диаграммын төрөл / баримтын төрөл ба харагдац

Харагдац	Бүтэц	Харьцаа
Статик харагдац	<ul style="list-style-type: none">•Класс диаграмм•Пакет диаграмм•Компонент диаграмм	<ul style="list-style-type: none">• Төлөвийн диаграмм• Хугацааны диаграмм• Үйл ажиллагааны диаграмм
Суурилуулалтын харагдац	<ul style="list-style-type: none">•Түгээлтийн диаграмм	<u>UML-д байхгүй</u>
Ажиллах үеийн харагдац	<ul style="list-style-type: none">•Объект диаграмм•Нийлмэл бүтцийн диаграмм	<ul style="list-style-type: none">•Явцын диаграмм•Холбоо харилцааны диаграмм•Харилцааны тойм диаграмм
Мэргэжлийн харагдац	<u>UML-д байхгүй</u>	<ul style="list-style-type: none">•Use Case диаграмм

Мэргэжлийн харагдац

- ПХ системийн хэрэгцээ шаардлага нь юу вэ?
- ПХ систем нь хэрхэн хамтын ажиллагаанд ашиглагдах вэ?
- ПХ эргономи буюу хэрэглээний тохь тухтай холбоотой асуултуудад хамаатай зүйлс энэ харагдацад хамаарна.

Мэргэжлийн харагдац

- **Зорилго**

Хэрэглээний салбарын хувьд хэрэгцээ шаардлагыг биелүүлэх загварыг аль болох техникийн шийдлээс ангид гаргах

- **Тодруулбал**

Системийн хэрэглээний бүрдэл хэсгүүд ба тэдгээрийн хоорондын зурвас болон холбоог загварчлах

- Үүнд хэрэглээний тохиолдлыг, хэрэглээний өгөгдлийг (өгөгдлийн загварын хэсэг) хэрэгжүүлнэ.

Бизнесийн процессыг мөн энэ харагдацад тодорхойлно.
(BPM)

Шаардлагатай бол ERD

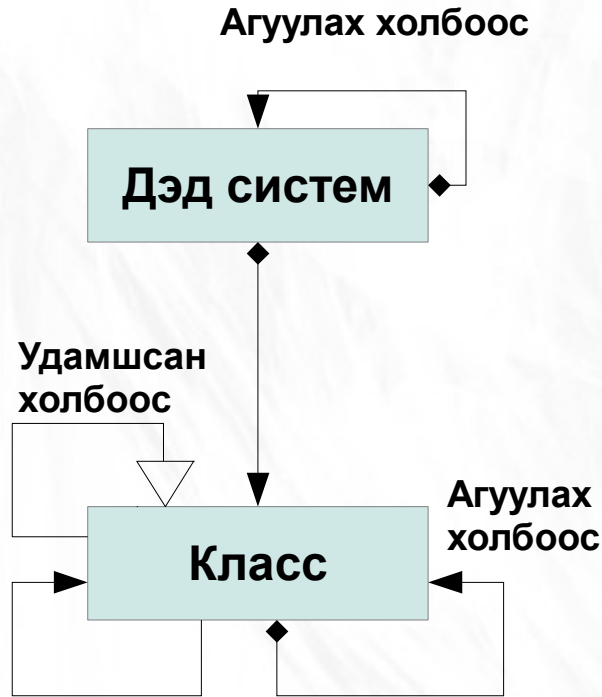
- **Хэл**

- Байгалын хэл, тодорхойлолтын хэл (ARIS)

Мэргэжлийн харагдац

- Үр дүн
 - Бизнес логик, процесс (BPM)
 - Үйлчилгээний төвшин (Service layer)
 - Хөрш системүүдийг орох гарах өгөгдлийн хамт дүрсэлсэн байх (blackbox)
 - Төсөлд оролцогчидтой ярилцах!

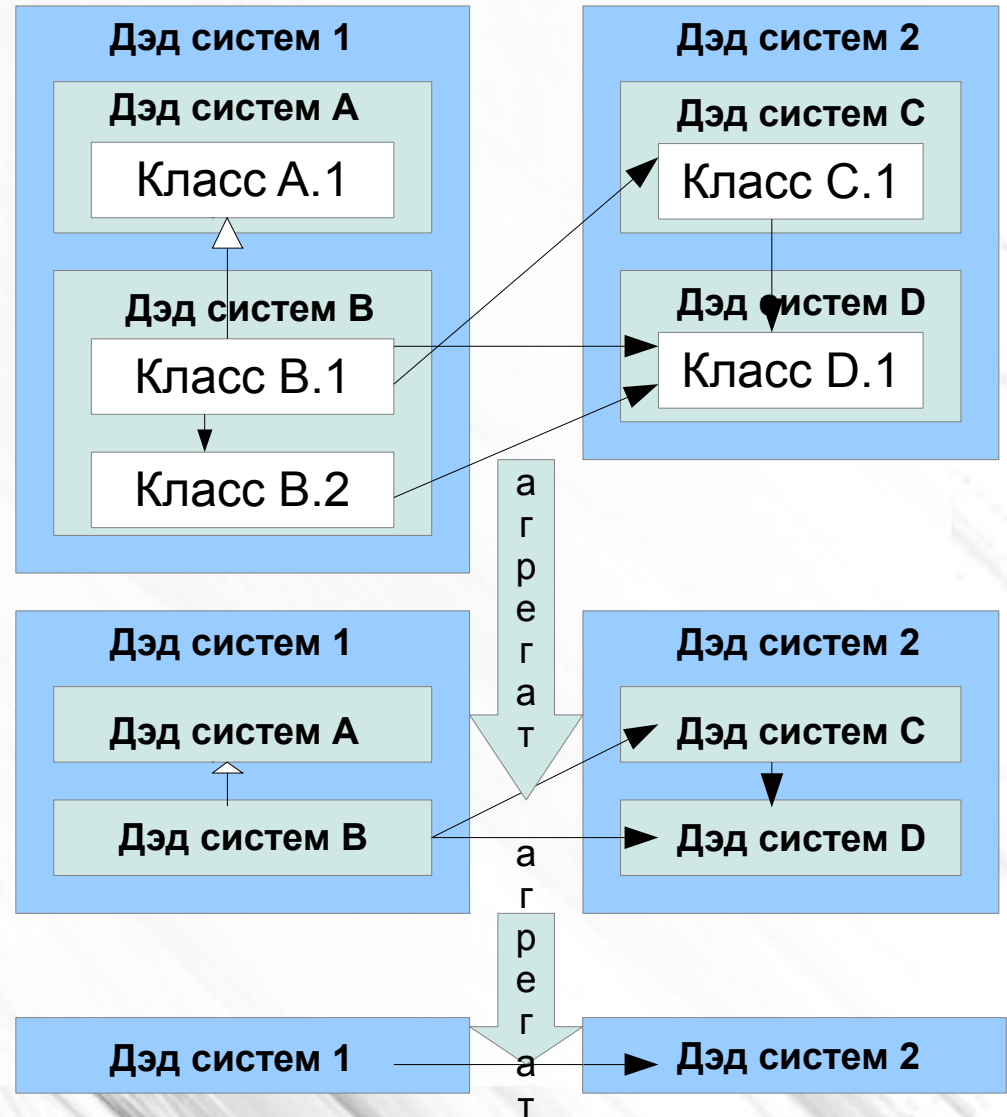
Статик харагдац



Хэрэглэх холбоос

Класс ба дэд систем нь өөр дотроо агуулсан элементээс Зурвас тогтооно.

Жава: Дэд систем = Packages



Статик харагдац

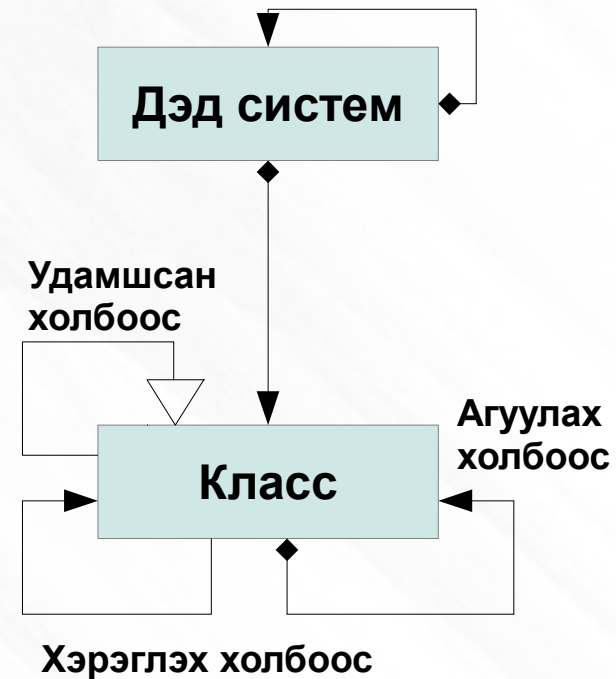
- Blackbox
 - Тодорхой хариуцлага эсвэл даалгавар
 - Импорт, экспорт зурвасууд
 - Чухал бүтээл кодууд
 - Тохируулах боломж
 - (Information hiding)
- Whitebox
 - Дотоод бүтцийг үзүүлсэн диаграмм
 - Дотоод элемент ба холбоо
 - Зохиомжийн загварууд

Статик харагдацыг бүтэцлэхэд юу туслах вэ?

- **Best practices**
- **Зохиомжийн загвар**
- **Архитектурын хэлбэр**



Агуулах холбоос



Хариуцлагаар зохиомжлох

Хариуцлагаар зохиомжлох (eng.: Responsibility-Driven Design) гэдэг нь 80-д оны эцсээр Ребесса Wirfs-Brock et.al.-н томъёолсон, зохиомжлох философи юм.

Objects are not just simple bundles of logic and data. They are responsible members of an object community.

Тохирох хэллэгүүд нь:

- **Seperation of concerns** (Dijkstra)
- **Modularity** (Parnas)
- **Cohesion** (Myers, Coad & Youdon)
- **Single Responsibility Principle** (SRP) (Robert C. Martin)

Класс бүр, пакет бүр, дэд систем бүр, төвшин бүр **тодорхой тогтоосон даалгаварт хариуцлагатай** байх хэрэгтэй!

Хариуцлагаар зохиомжлох

Лекц дээр жишээгээр нөхнө.

Том нэгжээс зайлсхийх

- Алдартай Bad Smell - “God Class”
 - Эх кодын 80-90%-г агуулсан
- Хэд хэдэн дэлгэц бүхий метод
- Хэдэн арван метод, хэдэн зуун example хувьсагч ба атрибуттай класс!

Тэнцүү хэмжээтэй класс, пакет, дэд систем
ба төвшинтэй зохиомжлох

Том нэгжээс зайлсхийх

- Класс 1000 хүртэл мөр
- Метод 100 хүртэл мөр
- Цикломатик нийлмэл байдал (Cyclomatic complexity) 10 хүртэл
 - (нэг процедураар үүсэх замын тоо)

Холбоосын төвшин

- Систем янз бүрийн төвшиндөө сул холбогдсон уу?
- Кодын аль хэсэг хэтэрхий их холбоостой байна вэ?



Bad smell: Галуун хүзүү

Холбоосын төвшин

- God class: Хэрэглэгчийн гадаргуун удирдлага, техникийн процессын удирдлага, өгөгдлийн сангийн холболт {*}Manager, {*}Service
- Холбоос: Удирдлагын класс 100-500 өөр класс хэрэглэж байх-> ООП биш процедурал зохиомж!

Хэмжээний харьцаа нь тэнцвэржилттэй байх нь сулавтар холбоог үүсгэдэг.

Зохиомжийн загвар

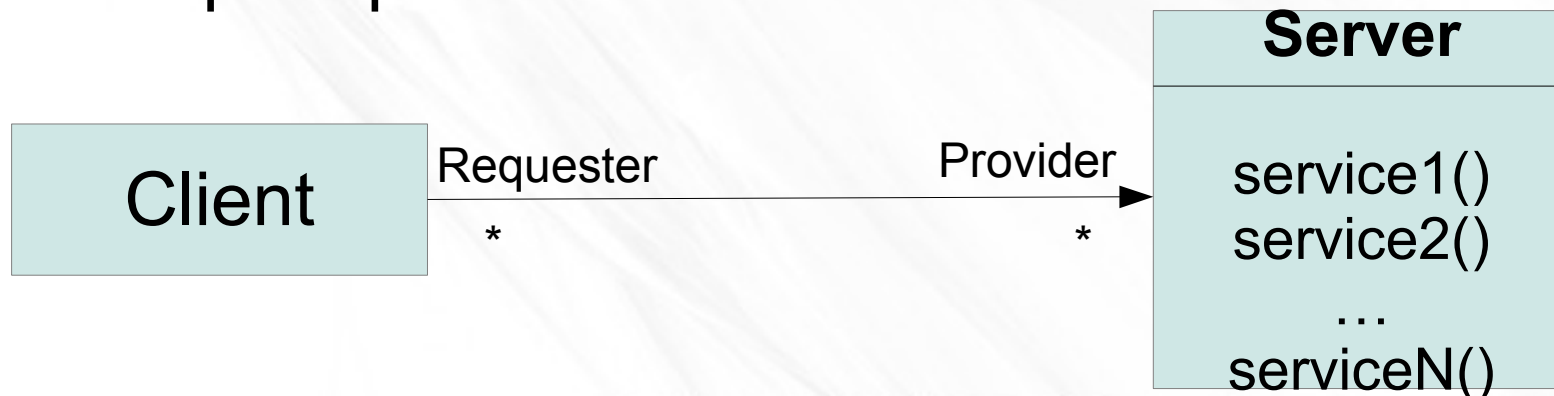
- Буддын сүмийн орой ба орос барилгын оройг санана уу.
- Амьдралд давтагддаг хэвшмэл асуудлын шилмэл шийдлүүд
- Зохиомжийн нэгжийн стандартчилагдсан эмх цэгц
- Локал хэрэглэгддэг
- Хэд хэдээрээ нэг системд хэрэглэгдэж болно.
- Ялгах нь
 - Үүсгэх загвар,
 - Бүтцийн загвар
 - Харьцааны загвар

Архитектурын хэлбэр

- ПХ системийн нийлбэр бүтэц!
- Хэвшмэл зохиомжийн асуудлууд
- Садан төрлийн бүтэцтэй домайн хамааралтай систем
- Архитектурын хэлбэр нь ПХ системд шууд, ямар нэг татгалзалгүй хэрэглэгддэг принципиал шийдлийн бүтэц!

Client/Server архитектур

- Үйлчилгээ үзүүлэгч дэд систем **Сервер** нь ерөнхийдөө хэрэглэгдэх функцууд ба үйлчилгээг санал болгоно.
- Нэг буюу хэд хэдэн үйлчлүүлэгчийн дэд систем болох **Клиентүүд** сервертэй харилцан ажиллана.
- Үйлчилгээний шаардлагууд нь сүлжээгээр тодорхой протоколлоор хийгдэнэ.
- Клиентүүдийн удирдлагууд нь өөр хоорондоо хамааралгүй



Архитектурын хэлбэр

- Архитектурын элементүүд
 - Ямар төрлийн элементүүд байна? Ямар ямар нэртэй байна?
 - Утга: Ямар утгата учиртай эсэх?
- Холбоо
 - Хэрэглэх, удамших холбоо г.м.
- Архитектурын дүрмүүд
 - Элемент тус бүр унтраалттай байж болох эсэх?
 - Элементүүд хэрхэн холбогдох вэ?
 - Элементүүд хэрхэн хамтран ажиллах вэ?

Ажиллах цагийн харагдац

Системийн ямар хэсгүүд ажиллах үед оршин байх болон хэрхэн хамтран ажиллах

- Систем хэрхэн эхлэх, ажиллах үеийн тохиргоо, захиргаа, системийн хяналт зэрэг аспектиудыг авч үзнэ.
 - Динамик, ажиллагааны харагдац хэмээн ч нэрлэдэг.

Суурилуулалтын харагдац

- Техник хангамжийн ямар компонентууд дээр ямар протоколлоор ажиллах системийн ажиллах орчинг тодорхойлно.
- Оролцогч элементүүдийн чадлын өгөгдөл, параметр зэргийг агуулна
- Үйлдлийн систем болон бусад гадаад системийг дүрсэлж болно
 - Дэд бүтцийн харагдац ч гэж нэрлэдэг.

Суурилуулалтын харагдац

- Топологи
 - Босоо хуваарилалт
 - Төвшингүүдийг ТХ дээр буулгах
 - Ж. нь N-tier системүүд
 - Хэвтээ хуваарилалт
 - Компонентүүдийг янз бүрийн компьютерууд дээр
 - Ж. нь EJB etc.

Дүгнэлт

- Архитектур = Элементүүд
+ тэдгээрийн шинж чанарууд
+ тэдгээрийн холбоо
- Архитектурын чухал харагдацууд
- Статик харагдацын бүтэц
 - Best practices
 - Зохиомжийн загвар
 - Архитектурын хэлбэр