\*\*Хүрхрээ аргачлалыг ашиглан системийг хөгжүүлэх нь: Онлайн банкны систем\*\*

\*\*1. Шаардлага цуглуулах:\*\*

- Төсөл нь банкны удирдлагууд, мэдээллийн технологийн мэргэжилтнүүд, эцсийн хэрэглэгчид зэрэг оролцогч талуудтай өргөн хэмжээний зөвлөлдөж, онлайн банкны системд тавигдах шаардлагуудыг тодорхойлж, баримтжуулах замаар эхэлнэ.

- Шаардлагууд нь хэрэглэгчийн баталгаажуулалт, дансны удирдлага, мөнгөн шилжүүлэг, төлбөрийн тооцоо, хуулга үүсгэх функцуудыг агуулна.

\*\*2. Системийн дизайн:\*\*

- Дээрх шаардлагад үндэслэн системийн архитекторууд болон дизайнерууд дизайны иж бүрэн зураг төслийг боловсруулах.

- Дизайн нь системийн архитектур, өгөгдлийн сангийн схем, хэрэглэгчийн интерфэйсийн зохион байгуулалт зэргийг багтаасан болно.

- Хэрэглэх технологи, хүрээ, програмчлалын хэлийг тодорхойлсон техникийн нарийвчилсан үзүүлэлтүүдийг боловсруулна.

\*\*3. Хэрэгжилт:\*\*

- Хөгжүүлэлтийн багууд системийн бүрэлдэхүүн хэсгүүдийг эцсийн дизайны техникийн дагуу хийнэ.

- Backend хөгжүүлэгчид өгөгдлийн сангийн бүтэц, бизнесийн логик, API төгсгөлийн цэгийг бий болгох чиглэлийг хариуцна.

- Frontend хөгжүүлэгчид өөр өөр модулиудад хэрэглэгчийн интерфэйсийг бий болгож, хэрэглэгчдэд саадгүй, ойлгомжтой байх боломжийг олгодог.

- Модуль бүр нь аливаа согог, зөрчлийг илрүүлэх, засахын тулд нарийн шалгалтанд хамрагддаг.

\*\*4. Туршилт:\*\*

- Туршилтын үе шат нь онлайн банкны системийн ажиллагаа, гүйцэтгэл, аюулгүй байдлыг баталгаажуулах олон шат дамжлагыг хамардаг.

- Туршилтууд нь нэгжийн туршилт, нэгтгэх туршилт, системийн туршилт, хэрэглэгчийн хүлээн авах тест (UAT) зэргийг хийдэг.

- Системийн функциональ шаардлага, мэдээллийн бүрэн бүтэн байдал, аюулгүй байдлын стандартад нийцэж байгаа эсэхийг шалгахын тулд туршилтын тохиолдлуудыг гүйцэтгэнэ.

\*\*5. Нэвтрүүлэлт:\*\*

- Систем туршилтын бүх үе шатыг давж, оролцогч талуудаас зөвшөөрөл авсны дараа түүнийг нэвтрүүлэхэд бэлтгэгдэнэ.

- Байрлуулалт нь системийг үйлдвэрлэлийн серверүүд дээр суулгах, тохиргоог тохируулах, хэрэв боломжтой бол хуучин системээс өгөгдлийг шилжүүлэх зэрэг орно.

- Байршуулсаны дараа системийн тогтвортой байдал, найдвартай байдлыг хангахын тулд өргөн хүрээтэй хяналт-шинжилгээ, нөөцлөх журмыг хэрэгжүүлдэг.

\*\*6. Засвар үйлчилгээ:\*\*

- Байршуулсаны дараа систем нь байнгын дэмжлэг, шинэчлэлтүүдээр хангагдсан засвар үйлчилгээний үе шатанд ордог.

- Хэрэглэгчийн санал хүсэлт, хөгжиж буй бизнесийн хэрэгцээнд тулгуурлан алдаа засах, гүйцэтгэлийг оновчтой болгох, функцийг сайжруулах зэрэг асуудлыг шийднэ.

- Системийг найдвартай, шинэчлэгдэхийн тулд хамгаалалтын засварууд, програм хангамжийн шинэчлэлт зэрэг тогтмол засвар үйлчилгээ хийдэг.

\*\*Дүгнэлт:\*\*

- Хүрхрээ аргачлал нь онлайн банкны системийг системтэй хөгжүүлэх, төслийн амьдралын мөчлөгийн үе шат бүрт нарийн төлөвлөлт, баримтжуулалт, туршилтыг баталгаажуулсан. Хэдийгээр уг хүрхрээ арга нь угаасаа хатуу ширүүн шинж чанартай хэдий ч оролцогч талуудад хөгжлийн үйл явцыг тодорхой байлгах, хяналт тавих боломжийг олгосон бөгөөд ингэснээр банкны шаардлага, харилцагчийн хүлээлтэд нийцсэн бат бөх, найдвартай системийг бий болгох юм.

\*\*Agile аргачлалыг ашиглан системийг хөгжүүлэх жишээ: Онлайн банкны систем\*\*

\*\*1. Sprint 1: Хэрэглэгчийн түүх ба спринт төлөвлөлт\*\*

- Хөгжүүлэгч, шалгагч, бүтээгдэхүүний эзэмшигч зэрэг Agile баг нь хэрэглэгчийн хэрэгцээ, бизнесийн зорилгод тулгуурлан хэрэглэгчийн түүхийг тодорхойлоход хамтран ажиллах.

- Хэрэглэгчийн түүхүүд нь хэрэглэгчийн баталгаажуулалт, дансны тойм, мөнгө шилжүүлэх зэрэг функцуудыг агуулах.

- Спринт төлөвлөлтийн хурал нь тэргүүлэх ач холбогдол, тооцоолсон хүчин чармайлт дээр үндэслэн эхний спринтэд ямар хэрэглэгчийн түүхийг оруулахыг тодорхойлох.

\*\*2. Sprint 2: Хөгжил ба өдөр тутмын зогсолт\*\*

- Хөгжил нь багийн гишүүн бүр өөрт өгөгдсөн хэрэглэгчийн түүх дээр ажиллахаас эхэлдэг.

- Өдөр бүр стэнд цуглаан хийж, ахиц дэвшил, саад бэрхшээл, тухайн өдрийн төлөвлөгөөний талаар ярилцдаг.

- Хөгжүүлэгчид давталттайгаар ажиллаж, хэрэглэгчийн түүхийг бөглөж, кодыг үндсэн салбар руу тасралтгүй нэгтгэдэг.

\*\*3. Sprint 3: Туршилт ба хянан үзэх\*\*

- Хөгжил дэвшлийн явцад шалгагчид аливаа согог, асуудлыг илрүүлэхийн тулд хэрэглэгчийн гүйцэтгэсэн түүхийг туршиж эхэлдэг.

- Тасралтгүй интеграци болон автоматжуулсан тест нь туршилтын үйл явцыг оновчтой болгож, алдааг эрт илрүүлэх боломжийг олгодог.

- Бүтээгдэхүүний эзэмшигч нь хэрэглэгчийн гүйцэтгэсэн түүхийг хянаж, багт санал хүсэлтээ өгч, хэрэглэгчийн хүлээлттэй нийцэж байгааг баталгаажуулдаг.

\*\*4. Sprint 4: Давталтын сайжруулалт ба санал хүсэлт\*\*

- Бүтээгдэхүүний эзэмшигч болон сонирхогч талуудын санал хүсэлт дээр үндэслэн баг одоо байгаа функцуудыг давтаж, шинэ функцуудыг нэмж өгдөг.

- Хэрэглэгчийн санал хүсэлтийг үзүүлэн, санал асуулга, ашиглах чадварыг шалгах замаар цуглуулж, цаашдын давталт, сайжруулалтыг удирдан чиглүүлдэг.

- Agile баг нь өөрчлөгдөж буй шаардлагад уян хатан, хариу үйлдэл үзүүлэх, шаардлагатай бол тэргүүлэх чиглэл, төлөвлөгөөг өөрчлөх чадвартай хэвээр байна.

\*\*5. Спринт 5: Байрлуулалт ба эргэн харах\*\*

- Спринт бүрийн төгсгөлд дууссан функцуудыг эцсийн туршилт, баталгаажуулалтад зориулж шатлалын орчинд байрлуулдаг.

- Agile багийнхан спринтийн талаар эргэцүүлэн бодох, юу сайн болсон, юуг сайжруулах, дараагийн спринт хийх арга хэмжээний талаар ярилцах зорилгоор ретроспектив уулзалт зохион байгуулдаг.

- Онцлогуудыг баталсны дараа тэдгээрийг үйлдвэрлэлд нэвтрүүлж, хэрэглэгчдэд хүртээмжтэй болгоно.

\*\*6. Тасралтгүй сайжруулалт ба ирээдүйн спринт\*\*

- Agile баг нь хэрэглэгчийн санал хүсэлтийг нэгтгэн, техникийн өр төлбөрийг шийдэж, дараагийн спринтуудад шинэ боломжуудыг нэмж, онлайн банкны системийг давтсаар байна.

- Тогтмол эргэн харах, санал хүсэлтийн гогцоонууд нь бизнесийн зорилтуудыг тасралтгүй сайжруулж, уялдуулах боломжийг олгодог.

- Agile аргачлал нь онлайн банкны системийг давталттайгаар хөгжүүлж, үйлчлүүлэгчдэд үнэ цэнийг хурдан бөгөөд дасан зохицох чадвартай болгохын зэрэгцээ өндөр чанар, хэрэглэгчийн сэтгэл ханамжийг хадгалах боломжийг олгодог.

\*\* DevOps аргачлалыг ашиглан систем хөгжүүлэх жишээ: Онлайн банкны систем\*\*

\*\*1. Дэд бүтцийг кодоор (IaC) тохируулах:\*\*

- DevOps баг нь онлайн банкны системд шаардлагатай дэд бүтцийг хангах, тохируулахын тулд дэд бүтцийг код (IaC) зарчим болгон ашигладаг.

- Сервер, өгөгдлийн сан, сүлжээний тохиргоо зэрэг дэд бүтцийн бүрэлдэхүүн хэсгүүдийг Terraform эсвэл AWS CloudFormation зэрэг хэрэгслүүдийг ашиглан программын дагуу тодорхойлж, удирддаг.

- IaC нь байршуулах явцад тууштай, давтагдах, өргөтгөх чадварыг баталгаажуулдаг.

\*\*2. Тасралтгүй интеграцийн (CI) дамжуулах хоолой:\*\*

- Хөгжүүлэгч баг нь кодын өөрчлөлтийг бүтээх, турших, нэгтгэх ажлыг автоматжуулах зорилгоор CI дамжуулах шугамыг бий болгодог.

- Хөгжүүлэгч нь хувилбарын хяналтын системд (жишээ нь, Git) код оруулах бүрд CI дамжуулах хоолой нь нэгжийн тест, интеграцийн туршилт, кодын чанарын шалгалт зэрэг хэд хэдэн туршилтуудыг автоматаар эхлүүлдэг.

- Автоматжуулсан бүтээц, туршилтын үр дүнг хөгжүүлэлтийн багт тайлагнаж, хурдан санал хүсэлт гаргах, асуудлыг эрт илрүүлэх боломжийг олгодог.

\*\*3. Тасралтгүй байршуулах (CD) дамжуулах хоолой:\*\*

- DevOps баг нь кодын өөрчлөлтийг үйлдвэрлэлд автоматжуулахын тулд CD дамжуулах хоолойг байгуулдаг.

- CI дамжуулах хоолой амжилттай дууссаны дараа CD дамжуулах хоолой нь шалгасан кодын өөрчлөлтийг дараагийн баталгаажуулалтад зориулж шатлалын орчинд автоматаар байрлуулдаг.

- Автомат байршуулах скриптүүд нь байршуулах үйл явцын тогтвортой байдал, найдвартай байдлыг хангаж, хүний ​​алдааны эрсдлийг бууруулдаг.

\*\*4. Хяналт ба ажиглалт:\*\*

- DevOps баг нь онлайн банкны системийн гүйцэтгэл, хүртээмж, аюулгүй байдлыг хянахын тулд найдвартай хяналт, ажиглалтын туршлагыг хэрэгжүүлдэг.

- Prometheus, Grafana, ELK стек зэрэг хэрэгслүүдийг бодит цаг хугацаанд хэмжигдэхүүн, бүртгэл, ул мөрийг цуглуулж, шинжлэхэд ашигладаг.

- Автоматжуулсан сэрэмжлүүлэг, мэдэгдлүүд нь системийн өндөр найдвартай байдал, ажиллах хугацааг баталгаажуулж, болзошгүй асуудлуудыг идэвхтэй илрүүлж, хариу өгөхөөр тохируулагдсан.

\*\*5. Дэд бүтцийн код (IaC) шинэчлэлт:\*\*

- Онлайн банкны систем хөгжихийн хэрээр DevOps баг IaC зарчмуудыг ашиглан дэд бүтцийн бүрэлдэхүүн хэсгүүдийн өөрчлөлтийг удирддаг.

- Нөөцийг нэмэгдүүлэх, тохиргоог шинэчлэх, шинэ үйлчилгээ нэмэх зэрэг дэд бүтцийн өөрчлөлтийг кодоор тодорхойлж, хэрэгжүүлдэг.

- Хувилбарын хяналтын системүүд нь дэд бүтцийн кодын өөрчлөлтийг хянаж, хамтран ажиллах, аудит хийх, буцаах чадварыг идэвхжүүлдэг.

\*\*6. Тасралтгүй сайжруулалт ба санал хүсэлт:\*\*

- DevOps баг нь хөгжил, үйл ажиллагаа болон бусад оролцогч талуудын хамтын ажиллагааг дэмжин, тасралтгүй сайжруулалт, санал хүсэлтийн соёлыг хөгжүүлдэг.

- Бэрхшээл тохиолдсон тохиолдлын талаар эргэцүүлэн бодох, сайжруулах шаардлагатай чиглэлүүдийг тодорхойлох, засч залруулах арга хэмжээг хэрэгжүүлэх зорилгоор эргэн харах уулзалт, үхлийн шинжилгээний ажлыг тогтмол зохион байгуулдаг.

- Хэрэглэгчдийн санал хүсэлтийн гогцоо, хяналтын систем, автоматжуулсан тестүүд нь онлайн банкны системд давтагдах сайжруулалт, оновчтой байдлыг бий болгодог.

\*\*Дүгнэлт:\*\*

- DevOps аргачлал нь онлайн банкны системийг тасралтгүй нэгтгэх, хүргэх, байршуулах боломжийг олгодог бөгөөд үүний үр дүнд зах зээлд гарах хугацааг хурдан, найдвартай байдлыг сайжруулж, өргөтгөх чадварыг сайжруулдаг. Дэд бүтцийн хангамж, кодын байршуулалт, хяналтын үйл явцыг автоматжуулснаар DevOps арга нь багуудад өндөр чанартай програм хангамжийг хурдан бөгөөд хариу үйлдэл үзүүлэх боломжийг олгож, хэрэглэгчид болон оролцогч талуудын хувьсан өөрчлөгдөж буй хэрэгцээг хангадаг.