

Manage a Repair Service Standard Assessment

Public beta
October 2018

Product and Data - 8, 12, 13, 15

Assessor: Tom Harrison (Lewisham Homes Digital Programme Manager)

Delivery - 2, 3, 4, 6

Assessor: Cate McLaurin (Hackney Head of Delivery)

User Needs / UX Design - 1, 10, 11, 14

Assessor: Susan McFarland-Lyons (Hackney Senior Delivery Manager)

Technical - 5, 7, 9

Assessor: Rashmi Shetty (Hackney Lead Developer), Dennis Robinson (Hackney Delivery Lead)

1. Research to develop deep knowledge of who the service users are and what that means for the design of the service.	Met
--	-----

The team creating the service should have a good understanding of user needs that has come from observing and engaging with end users, understand what users are trying to do when they engage with the current service (the user context, whether currently digital or not) and they understand the user needs – not just functional requirements – that the service will have to achieve in order to be successful.

Background

- **Discovery contextual research** and workshops with teams across the Housing Repairs service (Direct Labour Organisation operatives, Planners, Surveyors, Complaints, Repairs Contact Centre, Property and Asset Management, Case Management) to ascertain opportunities for improvement. Research techniques included:
 - **Observation of the teams in their natural environment**, completing everyday scenarios using existing technology to understand pain points and workarounds to **complete tasks**
 - **Qualitative interviews** to fully understand daily tasks, systems used, flow of data and pain points
 - **Journey mapping of existing workflow** to understand interactions between different teams and pain points
 - **Ideation workshop** to explore opportunities with teams across the Hackney Repairs service
- **Concept development** focused around the needs of the Leaks Hub Team:
 - 3 hour **journey mapping workshop** to understand the journey of a leaks case through the Leaks Hub team
 - 3 hour **design sketching session** to develop wireframes and simple design concepts
 - Creation of an initial set of **user stories** with the Leaks Hub Team
 - **User testing of early prototypes** - linked works order and floor plans
 - **User testing of hi-fidelity designs** for the works order prototype
- **User testing of the MVP** throughout its development cycle with the Leaks Hub and other teams that it could be relevant to including Complaints, Repairs Contact Centre and Surveyors

Key findings

How might we ...

How might we...
**improve visibility of
repairs job information**
for repairs staff and
residents?

How might we...
ensure that **reliable and
accurate data** is being
recorded and shared?

How might we...
provide staff with
**accurate asset details
and history** at a time
when they need it?

Recommendations

- Deploy and test application with other relevant teams in the Housing Repairs Service to gain a greater understanding of user needs.
- Further concept development to extend MVP functionality
- Focus on tenants' needs (particularly emotional needs) and consider how these could be addressed.

2. Ensure a suitably skilled, sustainable multidisciplinary team, led by a senior service manager with decision making responsibility, can design, build and improve the service.	Met
--	-----

The team should be empowered to design a service that meets users' needs; should share best practice and ensure that all viewpoints are taken into consideration throughout the design, build, implementation and improvement of the service post go-live. The size and expertise of the team and the roles required during the development of the service should be flexible during each phase, but must always include the service manager who will run the service on a day to day basis.

Background

- The team composition and roles:

DISCOVERY/ALPHA:

- Original Product Owner - (Direct Labour Organisation (DLO) Planned Works Manager
- Product Owner - Head of Repairs - Changed as the discovery phase determined where the team could deliver most value.
- Service Designer (Unboxed)
- User Researcher (Unboxed)
- Developer - Rails (Unboxed)
- Developer - API (Hackney)
- Junior Developer - API (Hackney)
- Delivery Manager (Unboxed)
- Delivery Manager (Hackney)

MVP/BETA:

All of the above, plus:

- Technical Architect - Rails (Unboxed)
- 2x Junior Developers - Rails (Unboxed)
- Lead Developer - API (MadeTech)

Hackney Governance

- Corporate Information Team
 - Technical and Security Architects
 - Universal Housing System Owner
 - Application Support Team
- The project initially focussed on working with the DLO. Discovery highlighted a different team who focussed on the most complicated repair cases - if the needs of this team were addressed, it would provide a service that would provide wider benefits across multiple teams at Hackney. Head of Repairs became PO for the project.
 - As the project went on, PO role became a PO team, Head of Repairs, the Leaks Hub Manager and a Leaks Hub Officer. This meant the PO team were able to give grass-roots context to which user stories would provide the most value in the MVP and ensure the team were empowered.
 - Ensuring Hackney API developers and Unboxed Rails developers worked as a single

cross-functional team on each user story ensured the most valuable functionality was delivered earliest and the process didn't fall into a waterfall process with integration bottlenecks.

Key findings

- Having a fully engaged Delivery Manager from Hackney working with the Unboxed team worked very well when there were organisational constraints to work through.
- Having a single person as a Product Owner with intimate knowledge of the daily work of the users works best, and they need to be empowered
- Use of language is important - for instance 'sign off' of user stories has a different meaning in different contexts.
- We've created buy in from the business to taking an agile iterative approach. The project has grown internal skills in this area.

Recommendations

- It would be more effective if it was possible to work closer with the product owner separately to refine user stories, writing acceptance criteria and the backlog, so that sprint planning is more effective.
- Sprint planning sessions - it would be more efficient to have one PO attached to the project with regular check ins with a Service Manager - aligned to GDS product and service owner roles
- More daily direct access to the users / PO, ideally co-location, would increase work throughput and reduce communication barriers. We have a disjointed situation at the moment where we have to email the team (a few steps away) to ask for them to look at a user story, which can lead to protracted email conversations and inability to quickly sign off stories.
- Address IT issues across the organisation so everyone can access modern tools to do their work efficiently. Encourage use of modern tools to manage communication and work. This would promote working in the open and prevents inefficient and opaque channels such as email.

3. Use agile methods	Met
-----------------------------	-----

Create a service using the agile, iterative and user-centred methods set out in the Government Service Design Manual.

Background

- The original plan was to have two two-week discovery sprints then move into the build, however, it became apparent that the DLO wasn't the right service to benefit from an MVP so we pivoted to deliver to a team managing the most complex repair cases.
- A number of prototypes were built to help understand which features should sit within an MVP.

- The project has been run using a cut-down Scrum process, with pragmatism and delivery in mind.
- This has entailed two week sprints, a Show & Tell after each one, adapting our process with a retrospective at the end of each sprint, sprint planning sessions with the whole team and the Product Owner(s) and daily stand-ups.
- In the stand-ups we usually follow the traditional format, however, at least once a sprint, we “walked the board” to ensure that everyone understood the current status of each of the stories in-flight.
- The Discovery phase involved a significant amount of user research with many user types across the Repairs service, exploration of the Hackney technical landscape, user journey mapping and co-design sessions with users.
- During the build phase we continued with open ended user testing to understand how the MVP should be shaped in future sprints and had design input on all stories.

Key findings

- Face to face co-location is preferred.
- Working in the open produces best results - use tools that allow open working i.e. Slack, Trello etc.
- More involvement from the PO in MVP enhancements would enable us to spend more time understanding and refining the backlog, using backlog refinement/ pre-planning sessions, which would make the delivery and planning process much smoother.
- Use of term MVP needed explaining - going forwards we need to use language that describes what we’re doing now.
- Co location of the team in the depot was effective in that we were close to the users. However, it wasn’t possible to be co-located with the leaks team directly, and this coupled with difficulties using trello and slack meant that there were frustrations and delays.

Appropriate tools and techniques used:

- Appear.in for easy video conferencing
- Trello for work/user story management
- Hackney Google Drive for project collateral
- Show & Tells and Week Notes uploaded to [HackIT Delivers](#) G+ community.
- Appropriate prototyping tools to explore the value of MVP features prior to full implementation (e.g. Kumu, simple Rails app)
- Brown-paper user journeys, post-its and sharpies
- Team employed a strong agile rhythm utilising 2 week sprints that started with a planning session and ended with a Show and Tell and retro.

Recommendations

- Have a single P.O. who is able to give a larger proportion of time to the project, so that we are reducing wastage of time.
- Explore how to use the role of an embedded delivery manager effectively and the benefits of shadowing and learning.
- Strategic decision from Hackney re SaaS project and development tools i.e. purchasing Slack, Trello.

- Look at how we might use new google functionality to support collaborative working.

4. Iterate and improve regularly	Met
---	-----

Build a service that can be iterated and improved in response to user need and make sure you have the capacity, resources and technical flexibility to do so.

Background

- Retrospectives were held every sprint with a variety of formats, these enabled us to address issues, risks and process adjustments in a timely and effective manner.
- Team composition was adjusted quickly as required (for example Developer from another agency joining the team to speed up development of API's).
- Change of Product Owner as the focus for the MVP was better understood.
- Increased involvement from users in testing as more functionality was made available.
- User testing continued through the implementation to ensure new users stories were captured for additional future enhancements.
- Design thinking maintained throughout to ensure that user experience remained a key factor during implementation of user stories.
- We originally had a tab named building work orders but after showing to users we've combined this into another tab and used filters instead.
- Our initial prototypes didn't use gov.uk design patterns but after a good discussion at a show and tell we adopted them and used the Hackney design patterns to develop a consistent feel to the MVP.

Recommendations

- Add analytics (Google and Hotjar) to the MVP so that usage can be understood and can inform future enhancements.
- Add feedback functionality to the MVP so that users can feedback while they're using it.
- Consider exploring whether creating product teams rather than a centralised app support team would help to enhance and support the service.
- Have a permanent Product Owner for the service so that the business feel like empowered owners and continuity is maintained outside of ICT.

5. Evaluate appropriate tools and systems	Met
--	-----

Evaluate what tools and systems will be used to build, host, operate and measure the service, and how to procure them, looking to reuse existing technologies where possible.

Background

- The Hackney API is the appropriate place for the service to access universal housing data from. The API currently serves the Report a Repair service and has access to the Universal Housing database.
- The Universal Housing application support team are cautious (performance risks) to running more direct queries against the Universal Housing database and recommend the Hackney API is to use data from the data warehouse whenever possible. The data held in the data warehouse can be up to 24 hours old.
- Real time data is required for this service. Leaks hub operatives need to know if work orders are raised today.

Key findings

- The Hackney API provides the ability to query repair requests and raise work orders. We will be able to reuse this API if there is a need to develop this service to create work orders.
- Civica are able to create stored procedures in Universal Housing to insert data. The creation of notes is missing from any current API and is not served well by the Universal Housing Web Service.
- The “related repairs” are a graph structure like this A->B->C->D... and will be inefficient to store/query in a SQL database.

Appropriate tools and techniques used:

- Ruby on Rails - Commonly used web framework. Well understood by multiple vendors at Hackney. Provides a large amount of out-of-the-box functionality and security. Used by income collection.
- Heroku - Cloud based PAAS system used by other projects in Hackney. Heroku is the most fully featured RoR PAAS with minimal integration costs. Used by income collection.
- QuotaGuard - Required to access the Hackney API through the firewall. QuotaGuard is a proxy service that allows our cloud based servers to access the API via a static IP address.
- Repairs API
 - C#, ASP.NET core
 - Augment the existing API used by manage a repair.
 - Create reusable components.
 - Optimised for least impact on the Universal Housing database by querying the data warehouse for any data before 10pm the previous day and the live Universal Housing database for the rest.
 - Developed in-house at Hackney to retain domain expertise.
 - Provide an abstraction layer to Universal Housing so that UH can be replaced without external applications needing large changes.
- Active Directory/Oauth - Iterated from other Hackney projects with access control shifted into active directory self service model.
- Neo4j/GrapheneDB
 - Used to augment the the Universal Housing database with additional

indexing capabilities

- Stores new relations between Work Orders
- Stores ids/references/dates but no textual content to avoid data privacy concerns.
- Postgres DB - used to store the background task job queue
- S3 for import files
 - To load the 5M+ records into the graph database we have extracted the index data to .csv files in S3 for loading into the graph DB via background tasks
- Docker - the standard development environment for the RoR application and Hackney Repair API is configured in docker as it is composed of multiple components which are most easily maintained in an automated setup like docker-compose.
- Travis CI/CD to support graph database.
- Heroku pipeline for deployment from staging to production

Recommendations

- Use insight from this project to deliver a Hackney Playbook for best practice
- Work to improve Universal Housing performance and setting up appropriate environments for Universal Housing Testing and Development.
- Develop UH read replica instead of dual sql queries to get information.
- Improve performance by shipping the transaction logs to cloud.
- Put tools in place to do continuous security checks on the application

6. Evaluate user data and information

Partially Met

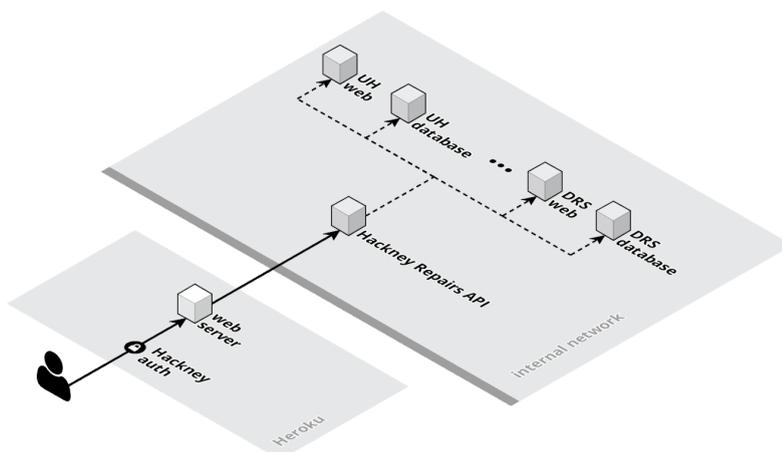
Evaluate what user data and information the digital service will be providing or storing and address the security level, legal responsibilities, privacy issues and risks associated with the service.

Background

- The MVP is not collecting or storing any new data (apart from the Graph DB which only stores anonymised work order references)
- The MVP shows data fields to users such as:
 - Work order details
 - Notes that might include email communication and sensitive comments regarding residents
 - Tenancy details / personal & contact details
 - Appointment details
 - PDF reports from visiting tradesmen
- The council has a responsibility to protect data under GDPR, the implementation of the MVP doesn't add any additional activities that need consideration with one possible exception....
- ...The front-end app is hosted on Heroku whose data centres are in Ireland, while the app doesn't store any of the information from the Hackney systems, it does consume and then present it.

Appropriate tools and techniques used:

- [Privacy Impact Assessment](#) undertaken in association with the Service Manager and Corporate Information and Knowledge Manager.
- None of the personal information shown in the MVP is hosted in a new system, the MVP is effectively a new piece of technology enabling officers to look at existing data to do their job more efficiently
- Authentication solution for access to service uses ADFS hosted on AWS (solution lifted from Income Collection project). Access can be granted on a user by user basis using an AD group.



Recommendations

- Address critical points raised in security assessment re API authentication and https

used end-to-end for API queries.

- Capture the rich learning about how data moves around Universal Housing and related systems so that future projects can benefit from it.

7. Use open standards	Partially Met
------------------------------	---------------

Use open standards, existing authoritative data and registers, and where possible make source code and service data open and reusable under appropriate licenses.

Background

Front end application:

- Code stored in <https://github.com/LBHackney-IT/repairs-management>
- RoR
- Postgres
- Neo4j
- GDS design system
- OAuth with Hackney active directory on azure

Hackney API:

- Uses some endpoints created by the raise a repair service
- Created new reusable endpoints
- Github link
- App Signal

Key Findings

We were able to reuse endpoints created in the raise a repair service, and we've been able to contribute new ones that can be reused by other Hackney Services.

Recommendations

- Team need to publish the code openly on GitHub before this standard is met.
- Use insight from the project to deliver a Hackney Playbook for best practice.
- Document the details of performance tools to handle exceptions, errors and application logging, how penetration testing was carried out and how Swagger json has been used, so that other teams can learn from the work.

8. Test the end-to-end service	Met
---------------------------------------	-----

Be able to test the end-to-end service in an environment similar to that of the live version, including all common browsers and devices.

Background

- Initial research focused on a broad understanding of the repairs service
- We took a holistic perspective – looking at repairs from the resident perspective, where a single repair can include multiple appointments and a number of operatives
- We prototyped with real data from the start – even the initial prototyping was done using real historical data from the service to ensure that all concepts were viable, feasible and desirable in a real-world context
- We included multiple stakeholders in our research, ensuring all perspectives and opinions were represented and heard – including the housing transformation team, financial services, complaints, data analysis, property and asset management, etc.
- We've done user testing in context with the live service to ensure that the MVP delivers real-world value. This included testing with browsers and environments that are used by staff on a daily basis, accessing our MVP through VMWare and legacy browsers, such as Internet Explorer.
- Acceptance Criteria were recorded with user stories as part of planning
- Tested with live/read data, the scale of testing done in terms of the number of live cases explored varied depending on the user story and was done in parallel with live Universal Housing

Key findings

- Including the usage of floor plans in the testing was important to understand the impact service could have in production. Complexity of the way leaks propagate could have led to a product that worked theoretically not being of real use once released.
- Testing with users in their context (current browsers and environments) was key to designing a product that will deliver value.

Appropriate tools and techniques used:

- Journey mapping:
we mapped out individual repair cases to understand a variety of possible cases and how they progress across multiple jobs and properties
- As-is service blueprint:
mapped from resident perspective and including interactions of all key teams (Repairs Contact Centre, planners, DLO operatives, surveyors, case management, external contractors) and systems
- Rails Application:
 - 2 testing environments are available on heroku. A user test data with access to the live Universal Housing database and data warehouse and a staging server with access to the dev Universal Housing database.

- Heroku pipelines allow us to deploy a copy of the application per github pull request for fine grained testing of specific features.
- Hackney API, 2 sandbox environments are available:
 - unboxedhackneyrepairs (access to live Universal Housing db and data warehouse)
 - unboxedhackneyrepairs_dev (dev Universal Housing db).
- Responsive design patterns (based on GOV.UK) are used to ensure compliance with standards and future proofing of the service, which might be accessed through mobile devices in the future
- Use of Graph database to join different repair jobs together helps to put together a picture of a repair holistically and better understand its journey

Recommendations

- Build appropriate Universal Housing test/dev environment that mirrors the Universal Housing production environment.
- Continue using a holistic approach throughout the repairs service, ensuring that any changes are well communicated with all teams – always considering potential negative impact of changes on other teams
- Further adaptations of layout to be used on mobile devices (e.g. iPads) to allow the MVP to be used by surveyors in the future

9. Make a plan for being offline	Met
---	-----

Make a plan for the event of the digital service being taken temporarily offline, and regularly test.

Background

- Users can revert to Universal Housing if necessary (in the event of the service going offline)
- We have a Basic Runbook
 - Check/diagnose heroku metrics page for platform errors.
 - Check/diagnose appsignal for application errors.
 - Restart “dynos” to return to service.
 - Trigger heroku maintenance mode if unable to return to service
- We have a Fix Runbook
 - Roll back to previous versions if outage caused by a new version
 - If analysis of metrics/appsignal reveal any errors:
 - apply a fix to the user-test system to verify fix
 - promote fix to production
- In the event of Data Corruption in Neo4j/Graphene
 - The graph database can be restored from the GrapheneDB backups webpage (Heroku->resources-tab -> GrapheneDB->backups-tab).
 - The data may fall behind for a while but the feed system will catch up.
 - GrapheneDB only keeps backups for 7 days. But a snapshot of the historical data can be found here:
<https://s3.eu-west-2.amazonaws.com/hackney-repairs-import/graph-backup-production-2018-10-17-1305-934mb.tar.gz>
- The service is hosted on a super resilient and fast cloud platform.

Key findings

- Publish the Runbook as part of Google Site being created for ongoing support of the service.
- Data is consumed from a combination of Hackney’s Data Warehouse and the Universal Housing Production database - this is a) a performance bottleneck and b) a data source that has little resilience and is prone to issues.

Recommendations

- Work to move a copy of the Universal Housing data to more resilient platform i.e. a cloud service. We think this would massively improve the resilience and reliability of this service.
- **Fallback plan needs to be planned.**

10. Make sure users succeed first time	Met
---	-----

Make sure that the service is simple enough that users succeed first time unaided.

Background

- **We involved the Leaks Hub team** throughout the development process ensured the MVP was built for them, with them
- **Developers were involved in user testing** sessions to gain a comprehensive understanding of user needs
- **We prioritised functionality and user stories** to ensure the MVP provided most value to the Leaks Hub team
- We used simple design patterns developed following the **GDS guidelines and design patterns** from the [gov.uk elements library](#)
- Collaboration with Hackney's Lead Front End Developer ensured designs were in **line with the evolving Hackney style library**
- High-fidelity **designs developed around daily tasks** of the Leaks Hub e.g. tabs for repairs history and notes
- **Testing of high-fidelity designs** to understand how the MVP should work
- **Iterative development**, based on feedback from user testing
- User testing with teams who are **less familiar with Universal Housing** to understand how they use the application and any obstacles to success

Key findings

Working closely with users throughout meant that we could ensure we were building a product that would deliver value and met user need.

Recommendation

- Use of gov.uk design patterns and user testing throughout means that accessibility is built in - however recommend specific accessibility testing is carried out to ensure it is fully accessible.
- Review the content to look at use of Universal Housing terminology. This may present a steep learning curve for some internal users. Consider contextual help or other ways to address this before rolling out to similar services.

11. Build a consistent user experience	Met
---	-----

Build a service consistent with the user experience of government digital services, including using common government platforms and the Government Service Manual design patterns.

Background

- Our approach throughout the process was following the **GDS service manual** around agile product development (iterative, user-centered)
- The application has been developed following the **GDS guidelines and design patterns** from the [gov.uk design system](#) (including extensive research into GDS admin interfaces) and Hackney style guide
- We also explored [design patterns published by Home Office Digital](#) particularly around admin interfaces to understand how we could best present all the information on one page. HMRC design was also useful as it has more of an admin interface specific approach.
- We engaged with **GDS mailing lists** - particularly for [Digital Service Designers](#) - to

- browse for design patterns
- We put effort into making sure the language on the application was similar to language that they are familiar with in Universal Housing

Key findings

Using existing design patterns and learning from other similar services enabled us to produce a consistent user experience.

Recommendations

Continue iterative user testing, consider installing user tracking & feedback tools such as Hotjar so that we can track how users use the service

Explore what design elements can be contributed back to the Hackney component library

12. Encourage everyone to use the digital service	Met
--	-----

Encourage maximum usage of the digital service (with assisted digital support if required).

Background

- We are currently focusing on Leaks Hub users, but roadmap includes work to adapt product for Repairs Contact Centre and roll out to those wider users
- Use of the new product is mandated for staff, but we are making sure that it's easy to switch from Universal Housing to MVP by including key users in concept development and testing
- Users have complained about eye strain after using Universal Housing for prolonged periods of time due to small font size
- Co-design – involving users in concept development and testing to increase buy in
- We involving senior leadership in development and feedback so they could adapt internal processes accordingly if needed

Key findings

- Even without additional training users find it easy to use the MVP and in fact are able to perform tasks faster than with Universal Housing (side-by-side comparison)
- Font size etc. follows GDS patterns making MVP much friendlier to users with accessibility needs
- Ensuring efficient performance so that users want to use the new system (and see value over using Universal Housing)

Recommendation

- Phased approach to introducing new users such as legal disrepair and complaints to ensure that additional usage doesn't make Universal Housing unstable
- Addition of functionality to make usable by e.g. Repairs Contact Centre or surveyors needs
- Spread the word via internal channels – HackIT community, Slack, show and tells – to ensure everyone is aware of potential use cases, including further development of the Hackney API

13. Identify performance indicators	Met
--	-----

Identify performance indicators for the service, incorporating existing indicators and publishing to a performance platform, if appropriate.

Background

We developed a range of KPIs for the services based on current performance data and service improvement goals. For instance we know that:

- A third of all plumbing and surveyor jobs deal with leaks, costing the council ca. £1M
- Leaks often involve multiple appointments, multiple trades and multiple properties, resulting in multiple chase-up calls from residents and complaints
- 6 Leaks Hub officers raised ca. 4,000 jobs per year
- It takes an officer on average ca. 8 min to process a task

We carried out data analysis of existing Universal Housing data, using Qlikview

Used Theory of change methodology – linking improvements to outcomes

Carried out a side-by-side comparison – filming of use of Universal Housing vs MVP

Key findings

We have improved the time it takes for Leaks Hub Officers to perform their tasks:

- 20-40% faster than Universal Housing
- 75% fewer clicks than Universal Housing

Less training is required on the new service due to more intuitive interface

For the Hackney repairs API we have:

- Doubled the number of endpoints
- Improved the technology
- More apps consuming the API

Ultimately this API can be used to provide housing and repairs data, rather than interfacing directly with Universal Housing.

Long-term goal is to increase residents' satisfaction with the repairs service and to reduce overall cost per leak. We can deliver this by:

- extending the use of the MVP to other teams (Repairs Contact Centre and Surveyors)
- integrating floor plans in MVP
- integrating additional asset info from Codeman (or other Property and Asset Management software)
- improving Out Of Hours repairs process
- improving data flow from external contractors

Recommendations

- Install analytics software, such as Google Analytics or Hotjar, to the application to gather performance data and monitor usage by the Leaks Hub Team
- Monitor impact on performance of Universal Housing database (liaising with IT)

infrastructure team)

- Analyse call handling time in Leaks Hub pre and post-release to ensure that reductions seen in test are realised in production.
- Use new ability to link jobs to see how long it takes to resolve problems from a resident's perspective before and after release to production.

14. Do ongoing user research	Met
-------------------------------------	-----

Put a process in place for ongoing user research, usability testing to continuously seek feedback from users, and collection of performance data to inform future improvement to the service.

Background

We plan to

- **Observe Leaks Hub Officers** using the MVP to understand how they navigate around the application, any problems / bugs they encounter and missing functionality
- **Continue to generate and test user stories** with the Leaks Hub Team and iterate based on feedback
- **Deploy and test the application with other teams** across the Hackney Repairs Service to understand how they use it and where there is missing functionality
- Install a feedback mechanism to the application to capture problems and ideas for improvement from the Leaks Hub Team
- Install analytics software, such as Google Analytics or Hotjar, to the application to gather performance data and monitor usage by the Leaks Hub Team

15. Test with senior manager	Met
-------------------------------------	-----

Test the service from beginning to end with appropriate council member or senior manager responsible for it.

Background

- Senior managers involved in the inception workshop at the beginning of the project
- Head of Repairs as the product owner from Hackney engaged throughout the project – testing and presenting the product to the wider service
- Product has been presented at the Housing Steering Group
- Senior Managers have been invited to Show and Tells which take place fortnightly

Key findings

Videoring the new service and existing process side by side was compelling and powerful evidence.

Recommendations

- Continue involvement of senior management
- Consider demoing the service to relevant Cabinet Member.