**Issue5**

Here, usability challenges will be defined as product features or situations in which the design and implementation of EHRs do not align with the cognitive and/or workflow requirements and preferences of users within and across professional and patient roles and settings.

------

Each of these individual offerings represents a partial solution to important usability challenges listed above. Leveraging multiple third-party solutions, i.e. these or others, within a clinical environment highlights the "plug and play" modularity that SHR empowers, demonstrating short term benefit to clinicians and patients without lengthy development timelines of a bespoke, potentially duplicative technology.

------

2) Support team-based care  
  
- Allow physicians, or AI-agents, to dynamically allocate and delegate work to appropriate members of the care team as permitted by institutional policies. |  
  
5) Reduce cognitive work load

------  
 Eva is an NLP-powered algorithm with access to the EHR and practice management software. She records her conversation with the patient within the EHR. Eva can report or collect structured data observations to or from the patient. When she does not know an answer, she triggers a medical assistant, triage nurse, or scheduler to call the patient, depending upon the issue. She can address most of the common reasons for calling the physician office, e.g. scheduling appointments, obtaining recent lab results, requesting updated referrals or medication refills, without a human-in-the-loop, and quickly trigger the appropriate human when necessary.

----------

- \_Product design versus implementation\_  
  
  Poor EHR design limits the potential for physician-patient engagement by interfering with face-to-face interaction with patients. Many local implementations of EHR products have been customized in an attempt to suit local needs, but the quality of these implementations varies greatly. EHR implementation is often viewed as a fixed cost, rather than an ongoing investment, requiring regular updates, training and iteration to function optimally.  
  
- \_Cognitive requirements versus workflow requirements\_

**Issue 22**

Signal wouldn't have to hide key verification and the contact information view would be accessible in a appropriate way. The overlay could state something like: [ Call Verify More ]

**Issue 1:**

## How will it help?  
  
- Instances can maintain bot driven logic without requiring a fork of core Rocket.Chat  
  
- Customised identity for bots in a Rocket.Chat instance distinct from the “system bot”  
  
- Clearer semantics between system notifications and instance’s specific business logic  
  
- Access to apps API endpoints will open new features to internal bots

**Issue17**

#### Goals of chat software  
  
- Be open and promote inclusivity  
  
- Reduce the need for emails  
  
- Enable access to conversations at any time on any device (not just while online)  
  
- Offer ability to manage conversations across projects and reduce the need for meetings.  
  
#### Useful features of Chat  
  
- Messaging and files  
  
- Send messages and comments across public, private, and 1:1 channels

**Issues 15**

- Some customised keyboard shortcut keys (we probably won't migrate the current settings but some keyboard shortcuts that relate to specific web pages can be supported if we have time)  
  
- Launching KeePass from within Firefox (launching executables is banned although if a user were to install a hypothetical "KeeFox helper" executable we might be able to get something to work but it is unlikely that will exist on most user's systems so this work goes to the bottom of the priority list)  
  
- Multi-page logins (the code that will deal with filling and saving passwords gets created and destroyed on every page load, making this a technically difficult problem to solve)  
  
- Some assistive UI (e.g. new user advice, tips, security warnings). Things like "Save/update password?" should be possible in some form but may suffer from reduced usability  
  
- Usage metrics (this is useful for planning how to improve KeeFox but with limited control over the changes we are making for version 2.0, a reduction in the number of platforms running version 2.0 and a lack of time to analyse and act on the limited data we could collect, it's not worthwhile initially)