TRADE-OFFS UNDER PRESSURE: HEURISTICS AND OBSERVATIONS OF TEAMS RESOLVING INTERNET SERVICE OUTAGES

John Allspaw

LUND UNIVERSITY
SWEDEN

Date of submission: 2015-09-07

TRADE-OFFS UNDER PRESSURE: HEURISTICS AND OBSERVATIONS OF TEAMS RESOLVING INTERNET SERVICE OUTAGES

Thesis/Project work submitted in partial fulfillment of the requirements for the MSc in Human Factors and System Safety

John Allspaw

Under supervision of supervisors Drs Anthony Smoker & Johan Bergström

Abstract

The increasing complexity of software applications and architectures in Internet services challenge the reasoning of operators tasked with diagnosing and resolving outages and degradations as they arise. Although a growing body of literature focuses on how failures can be prevented through more robust and fault-tolerant design of these systems, a dearth of research explores the cognitive challenges engineers face when those preventative designs fail and they are left to think and react to scenarios that hadn't been imagined.

This study explores what heuristics or rules-of-thumb engineers employ when faced with an outage or degradation scenario in a business-critical Internet service. A case study approach was used, focusing on an actual outage of functionality during a high period of buying activity on a popular online marketplace. Heuristics and other tacit knowledge were identified, and provide a promising avenue for both training and future interface design opportunities.

Three diagnostic heuristics were identified as being in use: a) initially look for correlation between the behaviour and any recent changes made in the software, b) upon finding no correlation with a software change, widen the search to any potential contributors imagined, and c) when choosing a diagnostic direction, reduce it by focusing on the one that most easily comes to mind, either because symptoms match those of a difficult-to-diagnose event in the past, or those of any recent events.

A fourth heuristic is coordinative in nature: when making changes to software in an effort to mitigate the untoward effects or to resolve the issue completely, rely on peer review of the changes more than automated testing (if at all.)

ACKNOWLEDGEMENTS

This research and thesis is the result of numerous people providing support and guidance and encouragement to me. I am indebted to all of them.

I must first thank Dr. Richard Cook, whose infectious curiosity has continually propelled me forward to make connections between fields and domains of expertise – I would not have considered joining this Master's program without his nudging and encouragement.

Dr. David Woods also deserves many thanks for his thoughtful guidance and encouragement, especially with respect to his insight into process tracing.

I must also thank Dr. Anthony Smoker, who has always brought me to question critically the concepts and models that I otherwise would have taken as objectively true. Professor Johan Bergström has always been an inspiring teacher who provided all the right challenges at all the right times, and deserves a special thanks for knowing when I'm not digging deep enough for better questions. I must also thank Dr. Sidney Dekker for his encouragement, patience, and 'thrown in the deep end' approach, which is sometimes what I need.

At Etsy, I am incredibly grateful for the support that Chad Dickerson has given me in order to be a part of this program and lead a world-class engineering team at the same time. I also must thank each and every one of the participants in this thesis. You know who you are.

Finally, I must thank my wife Elizabeth and my children Jack and Sadie for all of the confidence, support, and inspiration that they gave me during this journey.

Thank you all.

TABLE OF CONTENTS

Abstract 3
Acknowledgements 4
Table of contents 5
List of tables and figures 7
Glossary 8
Background 10
  Introduction 10
  What Makes This Difficult: An Opaque Operating Theatre 10
Literature Review 13
  Theoretical foundations and existing literature to "think with" 13
  A Gap Found: software, software, everywhere, but not a human to think 20
  Synthesis 21
  Thesis Question 22
Research Design 23
  Study Overview 23
  Process Tracing 24
  Materials, Data Sources, and Procedures 27
  Generalizability, Bias, and Validity 29
Event Description 31
  Context of the event 31
  Event Narrative: Degradation of Performance in Signed-In Homepage 31
Results 37
  Coding Schema: Listening To The World 37
  Timeline Overview and Map 37
  Coordination episodes 39
  Diagnostics and Hypothesis Generation 45
  Disturbance management and tracking the state of the response 51
Analysis 52
  Levels of participation 52
  Findings in Coordination Episodes 52
  Identified heuristic usage and anomaly response patterns 53
Discussion 63
Conclusions 66
Appendix A – Initial coding schema 67
Appendix B – Coordinative Episodes Coding Schema 70
Appendix C – Participant Timelines 71
Appendix D – Screenshots of Dashboards 75

Appendix E – Tools used in analysis 82
References... 83

LIST OF TABLES AND FIGURES

Figure 1 - Schematic of anomaly-driven cognitive activities or lines of reasoning involved in the disturbance management cognitive task (reprinted with permission) (Woods, 1995) 16
Figure 2 - Illustration of the methodology used 23
Figure 3 - High-level overview of the transformation of behavioural and verbal protocols into findings 29
Figure 4 - Median and 90th percentile performance of the Member_Homepage_Sidebar API call 32
Figure 5 - Signed-in homepage with sidebar components 33
Figure 6 - Screenshot of log analysis shared by engineer, indicating the API method exhibiting the errors 34
Figure 7 - High-level overview of the systems architecture and typical request-response flow..... 35
Figure 9 - Participants refreshing signed-in homepage 39
Figure 11 - Coordination Episode 1 41
Figure 12 - Coordination Episode 2 43
Figure 13 - Coordination Episode 3 44
Figure 14 - Coordination Episode 4 45
Figure 15 - Coordination Episode 5 45
Figure 17 - Vertical lines marking code deploys on a time-series plot of errors 55
Figure 18 - Survey results: waiting for automated tests to finish 61
Figure 19 - Infrastructure Engineer 1 timeline 71
Figure 20 - Infrastructure Engineer 2 timeline 71
Figure 21 - Infrastructure Engineer 3 timeline 72
Figure 22 - Infrastructure Engineer 4 timeline 72
Figure 23 - Infrastructure Engineer 5 timeline 72
Figure 24 - Product Engineer 1 timeline 73
Figure 25 - Product Engineer 2 timeline 73
Figure 26 - Product Engineer 3 timeline 74
Figure 27 - Forums dashboard 75
Figure 28 - Signed-in homepage dashboard 76
Figure 29 - API Component Performance Dashboard Mode Menu 77
Figure 30 - Time period selection for dashboards 77
Figure 31 - The Public, Member, and Shop modes of the API component performance dashboard 78
Figure 32 - Portion of the deploy dashboard 79
Figure 33 - API caching dashboard 80

Table 1 - Participant breakdown by group and experience 27
Table 2 - Initial coding schema 67
Table 3 - Intercoder Percentage Agreement 68

GLOSSARY

Activity feed – A feature on Etsy's signed-in homepage experience showing a continually updating feed of activity that includes new items from shops or favorites from people that you "follow" on Etsy.

Anomaly response – The collective activities of recognizing, reasoning, and responding to signals produced by a disturbance detected in an otherwise normalized process or state.

API – (Application Programming Interface) is a set of routines, protocols, and tools for building software applications. A website typically uses one or more APIs, typically will be created to accomplish its business purposes. APIs include software routines that can be "called" to accomplish specific functions many of which are highly specialized.

CDN – A system of distributed servers (network) that deliver webpages and other web content to a user based on the geographic locations of the user, the origin of the webpage and a content delivery server. CDN services are typically used to improve the performance of web content via caching or other acceleration techniques.

Deploy, 'code deploy', or 'push' – An explicit change of software code in the production Etsy.com website or APIs.

Frozen shop – A shop on Etsy whose selling status is temporarily paused or otherwise disabled while a policy violation, late bill collection, or other terms of use issue is being investigated.

Heuristic -- An approach to problem solving, learning, or discovery that employs a practical methodology not guaranteed to be optimal or perfect, but sufficient for the immediate goals. Where finding an optimal solution is impossible or impractical, heuristic methods can be used to speed up the process of finding a satisfactory solution. Examples of heuristics are 'rules of thumb', educated guesses, and intuitive judgements.

HOLD – A message displayed in code deployment IRC queues at Etsy when there is an on-going event that necessitates preventing any new code changes not related to the event response.

From Etsy's internal wiki:

> "The push queues should be held whenever we're not confident that Etsy is in a stable place and we want to take a breather to get our confidence back.
>
> This could be because we are seeing errors, the API is behaving oddly, the apps are behaving oddly or for anything else where we need a break to investigate stuff."

IRC – Internet Relay Chat, an online chat service. Internet service companies typically will have engineers working in different locations, and a textual chat service is a common practice to allow them to communicate real-time during both low and high-tempo situations.

Princess – A web application environment intended to provide a last step in deployment, where engineers can see and use the production service with the new code changes "live" before deploying the new code to production.

Supergrep – An open-source log streaming and filtering application, written by Etsy. (https://github.com/etsy/supergrep)

#sysops -- A channel in Etsy's internal IRC specifically for communications about technical operations topics such as server or network architecture, operations practices, or configuration management.

Three-armed sweater/whoopsie – The general error page for Etsy.com, shown when a user's request results in an HTTP 404 status code.

Trust and Safety – Operating within the Membership and Community organization within Etsy, the Trust and Safety team provides customer service and policy enforcement in the marketplace.

Varnish – An open-source software application, primarily used as an HTTP accelerator designed for content-heavy dynamic web sites as well as heavily consumed APIs.

#warroom – A channel in Etsy's internal IRC specifically for communications during an outage, degradation, or otherwise adverse event.

Wordpress – An open-source software application generally known for hosting blogs.

Background

Introduction

Increasingly, business and government services are delivered via the Internet. In 2013, total e-commerce sales in the US alone reached $263.3 billion (Census.gov, 2013). Over ten thousand UK citizens use the Gov.uk website simultaneously ("Performance Activity on GOV.UK: Web Traffic," n.d.) on a daily basis at the rate of over 2 million sessions per day (Thornett, 2015), making use of digital services such as checking on immigration status, applying for driver's licenses, search employment policies, and ordering birth certificates (Welcome to GOV.UK, 2015).

But any transient losses of large-scale Internet services such as those providing web search (e.g. Google), e-commerce (e.g. Amazon, Etsy), social media (e.g. Twitter, Facebook) and news (e.g. CNN.com, Aljazeera.net) can have far-reaching effects even beyond adversely affecting the health of their respective businesses. An increasing amount of modern society's daily life demands these services to be available globally, 24x7 (Margetts, 2015)

We need not look further than the technical issues surrounding the launch of the United States' online health insurance exchange, HealthCare.gov, to recognize that web services can become critical quite quickly to large populations of people (Ford, 2013) Even social media sites have been shown to bring strong influence to elections and regional policymaking. (Bruns, Highfield, & Burgess, 2013)

While the loss of these services have wide-reaching effects that cross geopolitical and economic lines that are beyond the purview of any of the individual companies providing these services, the detection and resolution of availability issues (which can range from minor anomalies to regional degradations of functionality and performance, to full global outages) is generally the responsibility of relatively small teams within those organizations. In other words: when large-scale services have outage events, they usually break in ways that affect a large population of users, and they are then restored to health by comparatively small team of operations engineers.

What Makes This Difficult: An Opaque Operating Theatre

The operating environment of Internet services contains many of the ingredients necessary for ambiguity and high consequences for mistakes in the diagnosis and response of an adverse unexpected event. The first is the systems are uniquely *opaque,* manifesting in a number of ways. For example:

- There are multiple layers of abstractions that hide the underlying complexity from operators. As an example, there are (currently) four layers of abstraction in the TCP/IP model (Internet protocol suite, 2013) and they all can carry traffic (the flow of data across the Internet) using multiple protocols each with their own fault-tolerance, consistency, and availability behaviours.
- Performance variability of a given network or node in the network is not transparent or readily available to the operator, so measurement requires a deliberate effort to target specific metrics. Network traffic has been described as a "turbulent river over a rugged landscape" (Veitch, Flandrin, Abry, Riedi, & Baraniuk, 2002)
- There is increasing interdependence between services offered, owned and operated by separate organizations. For example, *social login* is a method for authenticating a user via existing login information for social media sites (such as Facebook, LinkedIn, Google+, etc.) in lieu of creating another login. This method provides the convenience of using a "single sign-on" across many services, and its usage has been growing steadily. As of

2010, Facebook Connect was being used by over 250 million people (Van DeGrove, 2010). This means that in order to use one service, the authentication for that service depends entirely on another potentially unrelated (both geographically, technically, and commercially) service, owned and operated by a separate organization.

This opacity can hinder diagnosis when services experience degradations or outages, and complicates the monitoring of the health of services, specifically when operators are looking for leading indicators of trouble.

The second challenge is that the Internet is a *dynamically distributed* networking of computers and other devices, which span all national borders and policies. There is no central coordinating agent of the Internet, and therefore any route from one node to another is non-deterministic. This can make troubleshooting problematic due to the continually changing structure of the network and the availability of the nodes between nodes at any given time. In fact, the problem space and dynamics of distributed systems is vast enough to warrant its own field of research within computer science, known as *distributed computing* (Distributed computing, 2014).

The third is that typically in the organizations that build and operate systems on the Internet, teams are geographically distributed around the globe, communicating and coordinating virtually. There can be no "non-verbal" cues between team members as they cope with the complexity that confronts them. Coordinating work across the team is also complicated by time zones; some members of the team are asleep when others are doing work under normal operating conditions.

A fourth challenge in this environment is that it is an *open* system: it is continuously interacting with content and information consumers and producers. This lends itself to complex system characteristics more so than a closed network of computers.

In fact, all of the fundamental characteristics of complex adaptive systems can be found in Internet services: (Miller & Page, 2009)

- Connectedness (in the nodes, routes, users, servers, etc.)
- Diversity (in the functionality of websites, geographic and regional idiosyncrasies, etc.)
- Adaptation (in the fault-tolerance mechanisms of networks and applications, as well as in organization's capacity to respond quickly to adverse scenarios, etc.)
- Interdependence (in the myriad of ways that multiple services and underlying infrastructure are required for even one of them to be available.)

The assertion that collectively, Internet services (or "cyberspace") comprise a complex adaptive system (CAS) has also been made from a defence and military context as well (Grisogono, 2006; Phister, 2010). They cite a number of characteristics:

i. *Causality is complex and networked*: i.e. simple cause-effect relationships don't apply – there are many contributing causes and influences to any one outcome; and conversely, one action may lead to a multiplicity of consequences and effects
ii. *The number of plausible options is vast*: so it is not possible to optimize (in the sense of finding the one best solution in a reasonable amount of time),
iii. *System behaviour is coherent*: there are recurring patterns and trends, but
iv. *The system is not fixed*: the patterns and trends vary, for example, the 'rules' seem to keep changing – something that 'worked' yesterday may not do so tomorrow, and

v. *Predictability is reduced*: for a given action option it is not possible to accurately predict all its consequences, or for a desired set of outcomes it is not possible to determine precisely which actions will produce it.

Cook explains in an addendum to his paper "How Complex Systems Fail" in *Web Operations:*

> "**It will be difficult to tell what has failed.**
>
> Because complex systems involve tangled and shifting dependencies, the symptoms of failure are often nonspecific. This is a version of the notion of "action at a distance"; the consequences of failure are likely to be separated in time and space from the causes. When such a failure occurs, diagnosis may require very high levels of expertise to untangle the skein of components. This is problematic in the operational context because the people who are operating the system when it fails usually know little about the structure itself. " (Cook, 2010)

By describing how "it will be difficult to tell what has failed" Cook alludes to the challenges people face in reasoning about complex system failures in order to do something about them.

In summary, software engineers find themselves in a very unenviable position when attempting to resolve an outage with their service.

Relevance

As discussed in the introduction, an ever-increasing amount of modern society depends on Internet services to be available and functional. Extended outages or degradations can have unintended consequences that can have far-reaching and surprising effects.

And yet, despite the complexity and opaqueness of these increasingly important services, when these services do degrade or experience outages, engineers respond successfully to them. Teams of engineers gather, communicate their observations and hypotheses, coordinate their actions and reach decisions, and ultimately bring the service back up.

This study will present a more specific form of these questions, but succinctly, the challenges that face engineers brings some fundamental questions to mind:

- How do engineers *think* about the outage they're facing?
- What makes them *good* at resolving issues when they do not have a set procedure for the scenario they find themselves in?

LITERATURE REVIEW

Theoretical foundations and existing literature to "think with"

From a *safety science* perspective, there is a dearth of research in the current literature that attempts to investigate the issue of team-based anomaly response in the domain of web and Internet engineering.

However, there is *not* a lack of human factors and systems safety research on the topics of teams engaging in understanding and resolving anomalies under high-tempo and high-consequence conditions. This research has been performed in other domains (healthcare, aviation, space operations, military, etc.) and patterns have emerged that would be prudent to explore.

In order to provide a firm foundation for this thesis, it is necessary to identify key theories and concepts that we can rely and build upon; they set the stage for the research to take place.

Teamwork and Team Knowledge

A single individual very rarely resolves most modern technical problems. This is because teams provide some advantages over individuals when it comes to detecting problems (Klein, 2006):

- They have a wider range of attention, allowing them to monitor more information channels. This provides an enormous advantage in being able to detect early signs of trouble.
- They can have a much broader range of expertise than individuals, enabling teams to detect more signals and patterns and appreciate the significance of more types of events.
- They have built-in variability. Whereas an individual can become fixated on an interpretation of a situation, a team is likely to have members who are not locked into the first interpretation and are able to represent alternative perspectives.
- They have a greater capability for reorganizing their activities. Thus, during periods of high alert, teams can shift resources to monitoring information channels. During periods of confusion about events, teams can create analysis cells to examine trends. Teams can dedicate individuals to monitoring communication links, free of other workload considerations.
- They can work in parallel, so that some members are acting on a current situational interpretation while others are engaged in sensemaking and still others are seeking information.

However, a team of people coming together to resolve a technical problem brings with it a number of challenges as well. More heads are not always better than one. Coordination and communication costs can adversely or positively change the progress of a successful resolution of an event:

> "If the strength of a team and organization is the ability to perform several tasks in parallel by distributing them among the members, then a limitation is the additional effort needed to synthesize the experiences of the parallel strands." (Klein, 2006)

This would indicate that there's more to successful teams than simply a shared goal or mutual knowledge. In a synthesis of seven different studies across a period of five years, Emily Patterson, David Woods, Nadine Sarter, and Jennifer Watts-Perotti weaved together patterns from their findings on how *cooperative cognition* takes place. (Patterson, Watts-Perotti, & Smith, 1998)

The patterns they found were:

1. breakdowns in coordination that are signaled by surprise,
2. escalations in activities following an unexpected event in a monitored process,
3. investments in shared understandings to facilitate effective communication,
4. local actors adapting original plans created by remote supervisors to cope with unexpected events,
5. calling in additional personnel when unexpected situations arise, and
6. functional distributions of cognitive processes during anomaly response.

These patterns shine a bright light on the various ways that teams can be effective by keeping abreast of pitfalls as an event unfolds and their attempts to resolve the issue change accordingly. Of particular interest to software operations organizations is the case when *functionally distributed* teams need to work together and potentially improvise in order to diagnose or take remedial actions on their way to a resolution of the incident, because in large-scale outage scenarios there are mixtures of generalists and specialists who have overlapping (but not identical) expertise.

Distributed Cognition

Even before we can look into how teams coordinate, we first need to grasp *where* this coordination happens. The theoretical framework of *distributed cognition* provides a way of understanding the collective ability of individuals applying their technical expertise to achieve a common goal (Hutchins, 1995; Neisser, 1976). This perspective asserts that cognition is out in the world, not just in the mind, and that the environment that in which cognition occurs (which includes artefacts used to communicate and coordinate around) has significant influence on how cognition occurs.

In order to study how teams coordinate, then, we must then take into account all of the actors in a given context: the humans (co-located and remotely located) and the technical artefacts (computers) that they are working with. (Maglio, Kandogan, & Haber, 2008) remark that "Collaborative troubleshooting involves coordinating activity and information from people and other sources." (p. 146) and that

> "Distributed cognition treats certain arrangements of people and artifacts as cognitive systems, effectively computing functions by transmitting representations (e.g., language, computer commands) across media (e.g., air, computer screens). The idea is that the cognitive computation can be (partially) understood by tracking propagation of representations in this way." (p. 146)

Cognitive Systems Engineering and Coordination in Joint Activity

Hutchins' work tells us *where* we should ask questions about teams and their work. Cognitive Systems Engineering goes deeper into what questions we might ask when we get there. Individuals (in many cases with different domain expertise) coming together to diagnose and respond to anomalies found in Internet services can be seen as an example of *joint activity*. The concept of joint activity brings with it some common characteristics and dynamics with respect to how individual agents communicate, cooperate, collaborate, and coordinate their information and behaviours. A canonical piece on this topic is (Klein, Feltovich, Bradshaw, & Woods, 2005) which outlines the requirements for joint activity to be:

- Interpredictability
- Common Ground
- Directability

Following quite naturally from Hutchins' assertion that cognition is situated "in the wild", Hollnagel and Woods unpack this idea further to say that in order to understand joint activity (and therefore *joint cognitive systems*) then we need to first understand (Hollnagel & Woods, 2005):

- "that cognition is distributed across multiple natural and artificial cognitive systems rather than being confined to a single individual;
- that cognition is part of a stream of activity rather than being confined to a short moment preceding a response to an event;
- that sets of active cognitive systems are embedded in a social environment or context that constrains their activities and provides resources;
- that the level of activity is not constant but has transitions and evolutions; and
- that almost all activity is aided by something or someone beyond the unit of the individual cognitive agent, i.e., by an artefact." (p.141)

The first point above aligns with Klein's observations about team versus individual problem detection, as well as the concept of distributed cognition.

Woods had brought what he termed dynamic fault management (Woods, 1995b; 1995a; Woods & Sarter, 2010) to the Cognitive Systems Engineering (CSE) approach to anomaly response.

Woods asserts that there is a collective pattern of reasoning in the cognitive activities that take place during anomaly response (Woods, 1995a):

- During **diagnosis**, people act on their best explanations of what is happening in order to plan their response
- **Response planning** includes taking corrective action(s) which are then monitored for success
- If they **recognize anomalies** in the course of monitoring, those observations are fed back into response planning
- **Safing** actions may be taken in order to contain or limit the anomalous behaviour and those actions can generate more information to use in response planning
- Any unexpected behaviour is then used to inform the **diagnosis** of what is happening

Woods' assertion was that there is no clear distinction between these activities; they can blur together. Response planning or safing can be seen as activities to support diagnosis or further anomaly detection, etc.

This set of activities continues as the response heads towards resolution. This process is depicted in Figure 1 (Woods & Hollnagel, 2006).

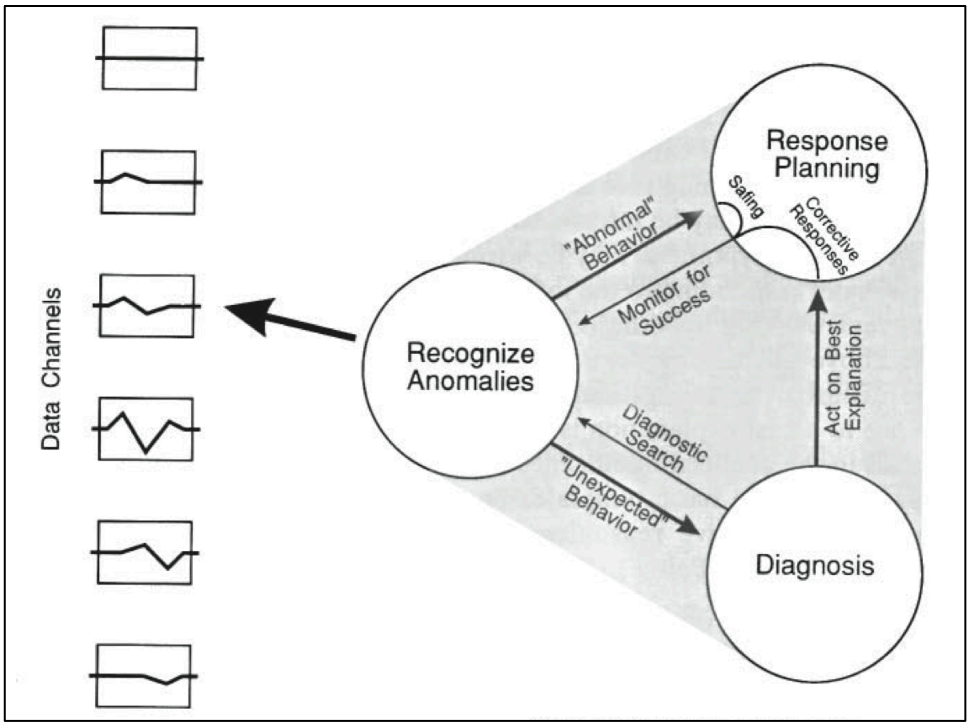


Figure 1 - Schematic of anomaly-driven cognitive activities or lines of reasoning involved in the disturbance management cognitive task (reprinted with permission) (Woods, 1995)

The same author expands on some of the challenges met within this pattern that can hamper the successful resolution of an unexpected event, especially in software-rich environments. Namely: data overload and underload (Patterson, 1999; Woods, Patterson, & Roth, 2002), signals provided by opaque and "clumsy" automation (Woods, 1999), as well as paradoxes involved with directed-attention and alarms (Woods, 1995b).

Perhaps the most relevant study to this thesis topic is one undertaken by Johannesen (Johannesen et al., 1994). The study explored how anaesthesiologists worked together during dynamic fault management conditions. The authors asked these research questions:

- How do they keep one another informed of relevant activities taken on the process or relevant assessments possessed by others?
- How do team members communicate their interpretations about the monitored process?
- How does this relate to the typical notion of explanation?

These questions and the resultant findings lay both the groundwork of theory and a framework for investigating teamwork applied to complex systems in any domain.

The closest research to date that resembles this topic in the domain of software operations is found in a book by researchers at IBM's Almaden Research Lab called *Taming Information Technology: Lessons From Studies of Systems Administrators* (Kandogan, Maglio, Haber, & Bailey, 2012)

Among the studies is one on the topic of what the authors call "crit-sit" (critical situation) events, and they take a qualitative approach that includes some of the data collection and analysis similar to what is done in this thesis (Haber, Kandogan, & Maglio, 2010). Although the event they explore does not focus on judgment heuristics or tacit knowledge, they bring attention to related research such as common ground and joint activity (Klein, Feltovich, Bradshaw, & Woods, 2005).

Computer-Mediated Communication

We cannot ignore the technical context of how communication takes place in this environment. While the cognitive and coordinative aspects of anomaly response in Internet software operations has not been studied, the topic of *computer-mediated communication* has, resulting in an excellent source of research: the *Journal of Computer-Mediated Communication.*

(Rogers, 1992) provides an example of how the use of computers in daily work can influence both the behaviour of multiple individuals on a team towards success and failure simultaneously, by improving some processes (e.g. assisting design in a virtual space) and introducing new problems to work around (e.g. the coordination of multiple people working on the same design at the same time.)

A study performed in a command-and-control military environment highlights some of the strengths and weaknesses for using *chat* as a medium of coordination during high-tempo scenarios (Knott, Nelson, & Brown, 2007).

We will need to take these strengths and weaknesses into account, as effective and clear communication is a requirement for continuing the process of *grounding.* Teams "lose" common ground for a number of reasons, identified by Klein, Armstrong, Woods, Gokulazschandra. Two of those reasons were that:

- They may experience an unexpected loss of communications or lack the skill at repairing this disruption;
- They may fail to monitor confirmation of messages and get confused over who knows what.

In either case, repair is required and that repair will bear costs. As the cost of this coordination and communication repair goes up to compete with the on-going process of anomaly response, teams have to rely on coping strategies.

Judgement Heuristics

In an environment where the amount of information available to explore (and rate at which it can be generated) greatly outpaces a team's ability to hold its collective attentional control, a number of coping strategies come into view. One of these strategies is to use *heuristics.*

Heuristics are a special type of *cognitive strategy* that allows decisions to be made (and actions to be taken) in a mentally economical way. They are 'short-cuts' that can be taken in order to avoid a more rigorous or thorough process of gathering potential options and evaluating each one of them for their appropriateness. (Kahneman & Tversky, 1974)

Daniel Kahneman and Amos Tversky's work is the most well known in this area of decision-making research, finding the use of heuristics as cognitive strategies employed when making judgements under uncertainty.

Nevertheless, Hollnagel argues that a number of *judgement heuristics* are at play in scenarios such as these, where multiple conflicting goals and uncertainty meet (Hollnagel, 2012):

- **Similarity matching** – judging the similarity between triggering conditions and stored attributes of appropriate action.
- **Frequency gambling** – choosing among partially matched options based on the frequency of their occurrence.
- **Representativeness** – to reduce uncertainty (if it looks like X, it is probably X)
- **Availability** – to reduce uncertainty (choosing by how easily the option comes to mind)
- **Focus gambling** - opportunistically changing from one hypothesis to another
- **Conservative focusing** - moving slowly and incrementally to build up a hypothesis
- **Simultaneous scanning** - trying out several hypotheses at the same time

Hollnagel points to these as strategies that are taken in order to reduce the complexity of a situation. Because these heuristics are shortcuts, they are a manifestation of the ETTO (efficiency-thoroughness trade-off) principle: they trade-off thoroughness with efficiency which can sometimes produce both upsides and downsides.

Focus gambling can be said to have such a downside. Dietrich Dörner (Dörner, 1980) termed *thematic vagabonding* as a pitfall that humans should watch out for as they attempt to cope with complex events:

> "People jump from one topic to the next, treating all superficially, in certain cases picking up topics dealt with earlier at a later time; they don't go beyond the surface with any topic and seldom finish with any." (Dörner, 1980)

Dörner adds to this a behaviour called *encystement*, or "cognitive fixation" where people can become fixated on a specific idea about what is going on or what solution is required, sometimes to the exclusion of incoming signals that may indicate otherwise. This would seem a pitfall to look out for an environment that can easily pair *data overload* issues alongside any ambiguity in the situation.

Dörner expands on a number of other potential biases and perspectives that people can take due to past experience, social and/or cultural power dynamics in the organization, etc. that manifest in which approaches they take towards explanation or resolution.

It would appear that in order to understand at a deep level how Internet engineers coordinate during the various phases of reasoning that entail resolving an outage scenario, it is important to look at how team members make use of these judgment heuristics while seeking to avoid some of the pitfalls.

Naturalistic Decision Making

Contrasted with decision-making research on heuristics and cognitive biases, the *naturalistic decision-making* (NDM) community tends to emphasize that skilled intuition comes from the recognition of cues found in a given situation that contain characteristics such as (Klein, 1993):

1. Ill-structured problems
2. Uncertain dynamic environments
3. Shifting, ill-defined, or competing goals
4. Action/feedback loops
5. Time stress
6. High stakes
7. Multiple players

8. Organizational goals and norms

This recognition comes from matching these cues with those stored in memory, and then evaluating a solution's appropriateness by mentally simulating if the solution will work well enough, if not optimally. Gary Klein has called this phenomenon *recognition-primed decision-making,* or RPD. (Klein, 1993)

Effectively, this is another form of mental shortcut. At first glance, RPD resembles the use of heuristics as defined and explored by Kahneman and Tversky: they are both cognitive strategies that people use in the face of uncertain or ambiguous situations requiring a solution or answer.

However, there are notable differences in the two perspectives and research communities, so they cannot be seen as equivalent. Kahneman and Klein have explored the similarities and differences of approach together.(Kaheman & Klein, 2009)

For the purposes of this thesis, a succinct way of treating this apparent fork-in-the-road of decision-making research can be found in Herb Simon's definition of *skilled intuition*: "The situation has provided a cue; this cue has given the expert access to information stored in memory, and the information provides the answer. Intuition is nothing more and nothing less than recognition." (Simon, 1992)

The key for this thesis is to recognize that both perspectives suggest that cognitive strategies used in situations that contain uncertainty (such as Internet service outages) are far from comprehensive, and will contain mental shortcuts. Whether these shortcuts are called "heuristics" or "skilled intuition" matters little; the goal is to identify them used in the wild.

Escalation

Finally, the concept of *escalation* (and recent research exploring it) must be taken into account since it is typically part of an outage response scenario. The Woodsian idea of an escalation is thus:

> "The concept of escalation concerns a process – how situations move from canonical or textbook to non-routine to exceptional. In that process, escalation captures a relationship – as problems cascade they produce an escalation of cognitive and coordinative demands which brings out the penalties of poor support for work. (Woods & Patterson, 2001, p. 291)"

This concept is later explored in a maritime (ship-bridge) simulation study which produced findings that supported an organization of data as an event unfolded: "For teams to successfully cope with a high data load in an escalating situation, there need to be processes to sort and prioritize rather than present all available data."

The result also showed "…that when the cognitive work in a team is focused on continuous sharing of as much data as possible, this can trap decision makers in a reactive behaviour during escalation." (Bergström, Dahlström, & Henriqson, 2010)

These results support the approach of exploring coping strategies (such as judgement heuristics) in order to find meaning in data that is ambiguous and at such a volume that considering all the data available would be impossible.

A Gap Found: software, software, everywhere, but not a human to think

The "dearth" of research (in this domain) on the topic of mentioned is not due to a lack of attention given to topics surrounding the availability, reliability, and resilience of inter-networked computer systems. On the contrary, there is a growing corpus of research in these areas.

Internet software operations and engineering (as a distinct discipline from computer science or even software engineering) is a rather new discipline, and its primary focus is the design and operation of Internet systems (Deshpande & Hansen, 2001). These systems can be seen to be *intractable* in a number of ways (Allspaw, Andersen, & Romanski, 2012) making anomaly response a challenging endeavour.

A significant amount of the literature that addresses anomaly or outage response in this domain focuses on the *prevention* and *detection* of problems, not the activity of recovering from them. The current body of work attends to topics such as:

- Distributed database systems performance and characteristics (Hellerstein & Stonebraker, 2005)
- Fault tolerance and consistency in distributed systems (Gilbert & Lynch, 2002)
- Scalability of systems (Chawathe & Brewer, 1998)
- Reliability of distributed systems (Ahmed & Wu, 2013)

These sources take the perspective that we should design our software and hardware in ways that are scalable, fault-tolerant, and as failure-proof as possible, and explore ways to do so. In other words: we should aspire to design and build *safe systems to begin with.* From a design perspective, this is a reasonable approach.

But it is incomplete with regards to safety. It is no more reasonable than stating that we can and should explore how to design safe aviation, power generation, or surgical trauma support systems. These domains have been researched from a safety science perspective, and have accepted that failures, unexpected outcomes, and accidents are not solely dependent on technical design of software, hardware, procedures and policies; they also depend on successful teams of humans to resolve issues as they arrive, even in escalating and uncertain scenarios. (Woods, 1999)

The issue is that the detection of anomalies and tolerance of transient faults and overall disaster recovery is generally approached from a *design* perspective whose scope is software and/or hardware. This can be posed as a question:

> "What is needed for the design of systems that *prevents* or *limits* catastrophic failure?"

This question has hitherto been explored via the domain of computer science, distributed computing and software engineering, resulting in architectural, algorithmic, and design pattern recommendations, and at this point has some rich history behind it, with the references cited above as starting points for the topics.

But of course this is a very different question than:

> "When our preventative designs fail us, what are ways that teams of operators successfully resolve those catastrophes?"

This thesis focuses on this area, and it represents bridging research from human factors and systems safety to Internet software engineering and operations.

Bridging Gaps. Filling Holes. Being Human.

Jens Rasmussen remarked in 1981 at an IEEE conference: "The operator's job is to make up for holes in designers' work." (Woods & Hollnagel, 2006). The "holes" Rasmussen refers to represent this gap between preventative design of software systems and the reality of complex systems failure (Cook, 1998). This research aims to gain understanding around *how* operators actually make up for those holes in practice, and specifically what heuristics or "rules-of-thumb" they employ in the process.

These "holes" that are being "made up for" are identified and recognized by experts, and this expertise is not always in the form of explicit knowledge or by using higher-order strategies to solve problems, but can be the result of tacit knowledge and subtle perceptual skills. (Klein & Hoffman, n.d.) This study focuses on exploring what *tacit* knowledge can be found in this practice.

(Woods & Hollnagel, 2006) relatedly posits the Law of Fluency as:

> "Well-adapted cognitive work occurs with a facility that belies the difficulty of resolving demands and balancing dilemmas. The adaptation process hides the factors and constraints that are being adapted to or around. Uncovering the constraints that fluent performance solves, and therefore seeing the limits of or threats to fluency, requires a contrast across perspectives."

Therefore, the challenge for this thesis is to uncover any adaptations that software engineers make in order to diagnose and resolve outages, focusing on any heuristics they might use.

Synthesis

The initial expectations for this research was threefold:

- To bring focus and sharper edges around how engineers perceive and navigate their way under uncertain scenarios.
- To understand what role *judgment heuristics* or *skilled intuition* play in how engineers keep common ground, communicate their perceptions to each other, and coordinate their actions to resolve the issue.
- To translate any findings into a form that can be used to progress the design of artifacts that aids operators in these uncertain situations.

This literature review provides a firm set of starting points and theoretical foundations for this study, expressed as:

- The field of software operations in Internet services can be seen through a similar lens as other high-tempo and high-consequence domains, due to the opaque nature of the environment and cognitive demands in play.
- Decisions made and actions taken by teams of engineers in an effort to cope with global-scale outages reflect an understanding that human decision-making and reasoning in

uncertainty is equally important as the design of the software aimed at preventing such events.

- While significant attention has been paid to the design of successful Internet services to include fault-tolerance and ways to survive *imagined* failure scenarios, the *unimagined* scenarios are what bring expertise to bear.
- Concepts such as anomaly response, disturbance and dynamic fault management, and common ground in joint activity provide useful frames of reference for exploring cognitive strategies that engineers may use.

Thesis Question

Ultimately, *human-human* coordination is at the heart of anomaly response in teams. This coordination happens amidst all of the members of a team

The research question then, is this:

> How do judgment heuristics influence Internet engineers' team coordination to resolve escalating large-scale outage scenarios?

Research Design

Study Overview

Methodology

In this study, the diagnosis and resolution of an outage in a global Internet service, Etsy.com, was explored in an effort to uncover which cognitive strategies (specifically, heuristics) are used by engineers as they work to bring the service back to a stable state.

Heuristics are "often used unintentionally, i.e., they have become second nature so that people rely on them and switch between them if results are not forthcoming quickly enough" (Hollnagel, 2009) – in other words, they can be seen to be *tacit* in nature. These 'rules of thumb' can therefore be hidden from both the practitioner and the researcher attempting to identify them. This is a challenge for this study.

This research is inherently *contextual*. Outages (and other anomaly response situations) are "complex, rich, and multifaceted" (Woods, 1993) events that occur in the world, not in a laboratory. The importance of context and naturalistic study is what drives the methodology.

In order to capture heuristics in this context, this study uses a case study approach, focusing on a specific outage as the case. The within-case exploration was performed using *process tracing* as a means of analysing engineers' "cognitive activities during complex work tasks" (Patrick & James, 2004). The data collected was then analysed with the goal of narrowing the focus to episodes where uncertainty could be identified via the in-situ utterances. These episodes could then be explored to bring to light any heuristics in play. This reduction and categorization was done via a cycle of open coding, content analysis, and semi-structured interviews.

A high-level and visual representation of this methodology can be seen in Figure 2.

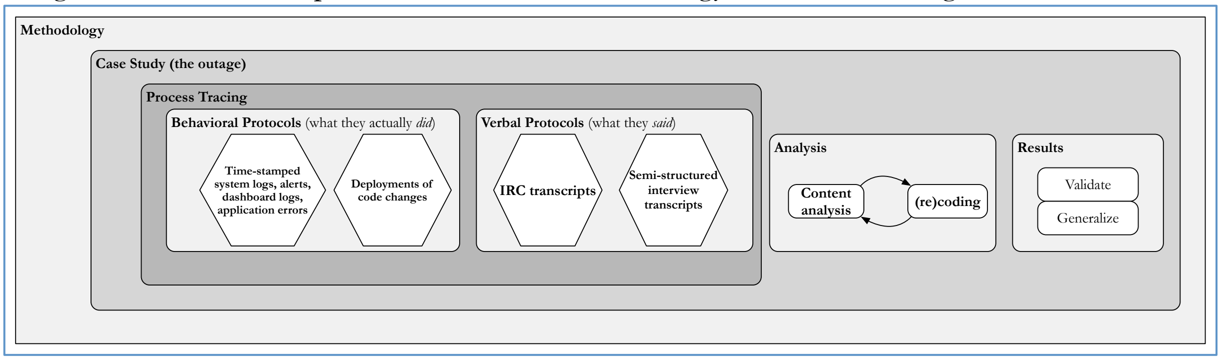

Methodology
Case Study (the outage)
Process Tracing
Behavioral Protocols (what they actually did)
Time-stamped system logs, alerts, dashboard logs, application errors
Deployments of code changes
Verbal Protocols (what they said)
IRC transcripts
Semi-structured interview transcripts
Analysis
Content analysis
(re)coding
Results
Validate
Generalize

Figure 2 - Illustration of the methodology used

Methods

A case study approach was chosen for this research, for a number of reasons. The first is that a case study fits a well-understood set of criteria in social science inquiry (Yin, 2013):

1. The main research questions are "how" or "why" questions;
2. a researcher has little or no control over behavioural events; and
3. the focus of study is a contemporary (as opposed to entirely historical) phenomenon.

The second reason is that because the focus of the study is to explore and identify what cognitive strategies (e.g., judgment heuristics), are being employed by a team of individuals during a *particular event* (an outage or disruption of an Internet service) it requires "developing an in-depth description and analysis" and "studying an event, a program, an activity, more than one individual" (Creswell, 2012).

(Creswell, 2012) effectively describes the rationale for how a case study is appropriate for this thesis:

> "Case study research is a qualitative approach in which the investigator explores a bounded system (a *case)* or multiple bounded systems (cases) over time, through detailed, in-depth data collection involving *multiple sources of information* (e.g., observations, interviews, audiovisual material, and documents and reports), and reports a case *description* and case-based themes." (p. 73)

Other qualitative approaches were considered, and dismissed as inappropriate based on the goal of the study. The goal was not to explore: any specific individual's experience (which would indicate using a *narrative* approach), the culture of the group under study (which would indicate an *ethnographic* approach), or theory development (which would indicate using a *grounded theory* approach). Therefore, the goal of the study very much directed the choice of qualitative research approach.

On how a case was chosen

Critical incidents (Flanagan, 1954) provide opportunities to explore cognitive processes of operators because they are both "pressurized" and "consequential" (Woods, D. and Cook, R.I., personal communication, July 2015).

This allows researchers to gain confidence that cognitive processes discovered are focused on the task at hand, and not extraneous activities. Flanagan explains what makes an incident critical in this way: "To be critical, an incident must occur in a situation where the purpose or intent of the act seems fairly clear to the observer and where its consequences are sufficiently definite to leave little doubt concerning its effects." (Flanagan, 1954)

The focus of this study is to explore the cognitive processes (namely, judgment heuristics) in play as engineers work to resolve an outage scenario. Heuristics in this case, are the 'rules of thumb' employed when engineers attempt to diagnose and respond to the anomaly when either uncertain or ambiguous signals exist to guide their decisions and actions.

Therefore, the outage event chosen for this study was driven by features that represented a high degree of uncertainty as perceived by the participants. While detection of the issue was not significantly problematic, initial signals perceived by engineers were not recognized as symptoms or part of a pattern that yielded insight into the underlying mechanisms.

Process Tracing

There are a number of well-worn methods available to the researcher looking to understand the cognitive processes of an operator, each with advantages and disadvantages.

Simply reviewing the behaviour of software systems will reveal nothing about the cognitive strategies employed by engineers overseeing it, and relying solely on practitioner's verbal reports about their experiences (let alone their cognitive strategies) will not provide reliable data, either. (Bainbridge, 1979; Ericsson & Simon, 1984; Nisbett & Wilson, 1977)

In order to infer what heuristics are being used, we can turn to a process-tracing methodology that lends itself well to disturbance management research, called *behavioural protocol analysis* (Woods, 1993). This approach makes use of many pieces of data and relies on artefacts (in this case, digital logs) to augment verbal reports. The pieces of data include:

a) direct observation of participant behavior,
b) traces of data acquisition sequences,
c) traces of actions taken on the underlying process,
d) records of the dynamic behavior of critical process variables,
e) records of verbal communication among team members or via formal communication media,
f) verbal reports made following the performance, and
g) commentaries on their behavior made by other domain knowledgeable observers. (Woods, 1993)

That data can then be "correlated and combined to produce a record of participant data acquisition, situation assessment, knowledge activation, expectations, intentions, and actions as the case unfolds over time." (Woods, 1993) which will provide a good contextual foundation on which semi-structured interviews (described in more detail below) could then triangulate on to explore how heuristics are being used. In other words, this data can be seen as *externalizations* of the mental processes. This approach is commonly called *process-tracing,* because the goal is to "map out how the incident unfolded including available cues actually noted by participants, and participants' interpretation in both the immediate and in the larger institutional and professional contexts." (Woods, 1993)

Process tracing techniques are "all oriented towards *externalizing internal processes or producing external signs that support inferences about internal workings.*" (Woods, 1993)

There are two main categories of verbal reports typically used in process tracing techniques: think-aloud or ("concurrent") protocols, and retrospective reports. In this study, retrospective verbal reports were taken, in the form of semi-structured interviews as part of a "cued recall" approach.

Potential pitfalls in verbal reports

(Schulte-Mecklenbeck, Kühberger, & Ranyard, 2011) cite some potential threats to validity in eliciting verbal reports retrospectively. The challenge is to minimize "justification bias"*,* which is the unconscious altering of their reports by ways of rationalization or other influences.(Huber & Seiser, 2001)

The authors assert, however that:

> "As long as participants are not warned in advance that a retrospective verbal report will be required, the procedure would not be reactive. However, if advance notice is given, a particular form of reactivity known as 'justification bias' may change the pre-decision process." (p. 119)

Therefore, care was taken in communicating about the interviews that were done with participants; they were unaware they would be asked about the specific event and of the format of the discussion in general.

Cued Recall

Process tracing in this study made use of *retrospective* verbal reports, since the outage in focus has already happened. Retrospective verbal reports are vulnerable to a number of threats to validity that need to be handled, and one of the ways this is done is through the use of "stimulated" or "cued" recall (Schulte-Mecklenbeck et al., 2011; Woods, 1993).

Stimulated recall involves 'replaying' an event with a participant by reviewing logs of actions and behaviours, and in-situ utterances in the timeline of the event, and probing their reasoning at critical junctures in the timeline via open-ended interview questions. These junctures are identified as relating to the study, which in this case is heuristic reasoning.

Normally, stimulated recall involves reviewing a recorded video sequence of activity with the participant. In this study, an amalgam of the behavioural protocols (system logs) and in-situ utterances (IRC transcripts) providing the narrative to be reviewed. Participants are confronted with their words and actions as they actually happened in the event, which frames the actions and potential decision points in the specific event and not in a hypothetical one.

Participants can then be asked about what brought them to make a particular comment or take a particular action, instead of how they would claim to act in a general sense in similar situations. The strength of cued recall in process tracing is the avoidance of solely relying on a participant's memory of an event.

The interviews were conducted with a similar approach to the Critical Decision Method (CDM), a *cognitive task analysis method* (Crandall, Klein, & Hoffman, 2006) intended to use deepening probes around a timeline in order to elicit tacit knowledge.

Each of the eight engineers was interviewed, with a mean interview time of 51 minutes. The interviews took place approximately six weeks after the event, and audio recordings of the interviews were sent to a transcription service. The transcriptions were then proofread before being coded.

The interviews were semi-structured, focusing on the timeline of externalizations found in the IRC transcripts (Figure 8) as well as logs indicating what dashboard web pages the engineer's browser requested during the event. In some cases, the dashboard contained many graphs, so the participant was asked to indicate which of the graphs on the page were of particular interest, and what they expected to learn from interpreting them.

Participants

Eight software engineers from varying groups within Etsy's engineering organization and experience worked together to resolve the issue. Three of the engineers are on *product* teams, which are largely focused on building and operating features on the website. Five of the engineers are on an *infrastructure* team, which is generally focused on "back-end" systems that store and serve data to many applications and features on the site. See Table 1.

Table 1 - Participant breakdown by group and experience

Participant	Years at Etsy	Years working as software engineer
InfraEng1	4	14
InfraEng2	2.5	9
InfraEng3	1	4
InfraEng4	4.5	17
InfraEng5	4	8
ProductEng1	3.5	10
ProductEng2	0.8	8
ProductEng3	2.5	13

Materials, Data Sources, and Procedures

The following sources of data were used to construct the process trace:

- IRC chat transcripts from multiple channels
- Access logs of what dashboards and graphs the engineers loaded into their browsers during the outage
- Timestamps of changes to the application code made during the outage
- Logs of any alerts (email and text message notifications) that triggered during the outage
- Semi-structured interviews with 8 engineers who participated in the event response

IRC transcripts

Internet software operations and engineering teams typically use multiple channels of communication; a predominant one is textual chat (Internet Relay Chat.) This element of the environment is interesting on a number of fronts:

- IRC is used mainly as a low-latency way to communicate amongst engineers, some of which are located remote from Etsy's main office
- Diagnosis and cross-checking of digital artefacts such as commands, logs, etc. make it a natural fit
- It provides a time-stamped running and persistent history
- One-to-many communication is the default

What IRC transcripts provide is a "written" record of how people communicated, coordinated, and cooperated during the event. These transcripts can give some insight into the *choreography* of the response (Klein et al., 2005) which can help identify critical points of decision-making and actions.

These transcripts can serve as the *externalizations* of engineer's cognition we are looking for as we construct and follow the process tracing, not unlike what might result from a concurrent or 'think-aloud' protocol, but without the prompting of a researcher.

Systems Logs

Time-stamped log files from technical systems can provide a valuable part of the trace necessary to construct in cognitive work studies (Wright & Monk, 1989). Logs were collected from Etsy's log servers that recorded data about various systems behaviours and engineer-driven actions during the event.

Logs are singular lines of text appended to a file, time-stamped to the second. The available logs in this study include the following, collected during the event:

- Webserver logs of what dashboards and graphs engineers requested from their browsers.
- Webserver logs of what features of the customer-support application engineers requested from their browsers.
- Webserver logs of the application(s) that was experiencing the outage, including both "access" and "error" logs.
- Deployment logs of any changes made to the application(s) experiencing the outage.
- Logs of any system-driven alert notifications sent to engineers.

Other artefacts also include screenshots that engineers shared with each other during the event.

Procedures

The study was conducted as follows:

a) A brief description of the initial research topic was given to the Engineering organization at Etsy.
b) Informed consent agreement was sought and received across a population of engineers.
c) A candidate outage occurred and was chosen as the case to study.
d) The following data was exported from Etsy's production servers and archived for further analysis:
e) IRC transcripts for 3 separate channels containing conversation relevant to the outage.
f) Access logs from dashboards, graphs, log analysis servers, and customer support applications.
g) Deployment logs of code changes being made immediately before and during the outage.
h) Logs of alerts sent (via email and text) immediately before and during the outage.
i) The IRC transcripts underwent an initial thematic open-coding process.
j) A pilot semi-structured interview was performed with one engineer, using a cued recall approach and open-ended questions.
k) Refinements to the coding scheme were made in order to further reduce the data and bring attention to potential heuristic usage
l) The remaining seven engineers were interviews in a semi-structured format, using cued recall.
m) The behavioral (systems logs), conversation (IRC transcripts), and verbal reports (interviews) were then analyzed by finding common themes and synthesized to make inferences about what heuristics were being used during the outage.

A visual representation of the process tracing procedure can be seen in Figure 3.

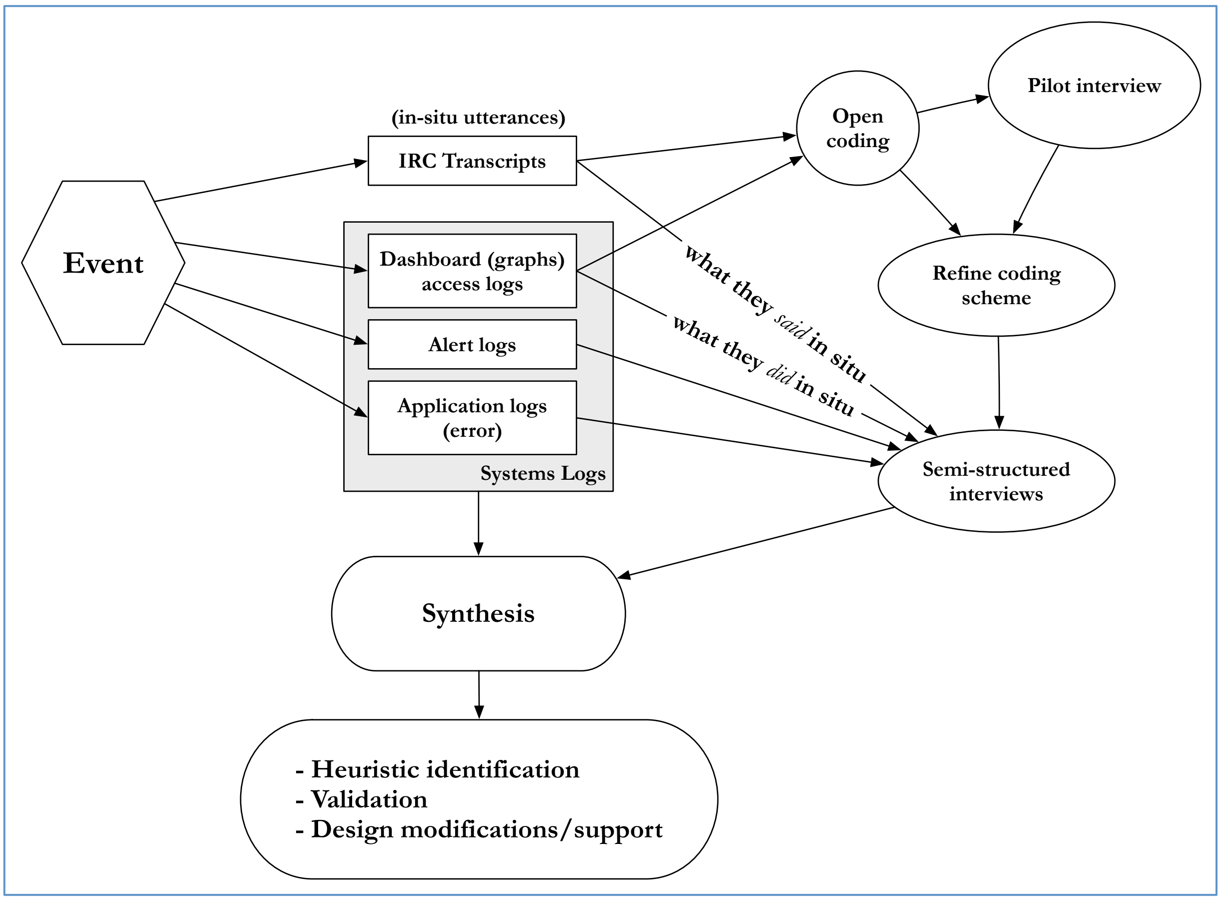


Figure 3 - High-level overview of the transformation of behavioural and verbal protocols into findings

Generalizability, Bias, and Validity

Generalizability

This study does not result in *statistical* generalizability, nor was it intended to. The focus in this research is *analytic* generalizability (Yin, 2013) in which constructs from foundational theory (synthesized from the literature review) can be interpreted and presented as represented by this case study.

Bias

This research used a case study of an Internet service outage. The criteria used when selecting participants for the study was twofold:

- They were part of the group that responded to the outage.
- They contributed significantly to the response. The volume of dialogue in the IRC transcripts largely determined significance.

The participants chosen represented 68.4% of the dialogue found in the #warroom IRC transcript.

The investigator was an "insider" researcher. This yielded a number of advantages: he was familiar with many of the technical details and argot involved in the case and experience with the practices of participants that a lay researcher would not have.

The researcher also had biases and opinions on:

- What behavioural data to collect and analyse.
- Which questions to ask during the interviews with participants.
- What tools to use in representing the process traces.
- How to interpret the process trace data and make inferences about heuristics identified.

Validity

In order to establish validity in this case study, a number of efforts will be made:

- **Establishing a chain of evidence.** This is critical for constructing a process trace to infer any cognitive activity.
- **Triangulation of data sources.** Participant externalizations cross-reflected with other participants can aid this validation.
- **Thematic refinement of coding systems.** Both collaborative coding as well as finding inter-coder reliability.
- **Review of the report draft by participants.**

EVENT DESCRIPTION

Context of the event

Etsy is an online "marketplace where people around the world connect, both online and offline, to make, sell and buy unique goods." As of Q22015, Etsy has approximately 32 million items for sale, 1.5 million active sellers, 21.7 million active buyers, and in 2014 had $1.93 billion in annual gross merchandise sales (About Etsy, n.d.)

December is a busy time for most ecommerce companies, due to the holiday gift-buying activity. In the United States, a marketing term for the Monday after Thanksgiving is "Cyber Monday" on which many online retailers promote buying activity. ("Cyber Monday," n.d.) The event in this study took place three days after Cyber Monday in 2014. Etsy reported that in the fourth quarter of 2014, "63% of active buyers as of December 31, 2014 logged in to our marketplace."

Event Narrative: Degradation of Performance in Signed-In Homepage

On December 4th, 2014, a portion of the Etsy.com website became unavailable. Normally, when members are logged-in, the homepage of the site has an "activity feed" that is personalized for them. This includes new listings for sale from shops that they have highlighted (or, "favorited") or bought from in the past, what their friends have favorited, recommendations based on past behaviour, and other items of content that are specific to the member. This feature was designed to make the experience easier for visitors to find what they want to buy, and is a particularly important feature to have during the holiday shopping season.

At 1:06pm Eastern Standard Time, reports of the personalized homepage having issues began appearing from multiple sources: the technical arm of the customer support group, a Product Manager for the Personalization Team, and others in the company. Instead of seeing the expected personalized homepage, a generic "trending items" feed was shown, which is not personalized and is the same for any visitor. This "trending items" feed is a contingency feature, shown only in the event that the personalized homepage cannot be constructed.

Coincidentally, the Personalization/Homepage Team (which includes product managers, designers, and engineers) were gathered into a conference room in Etsy's Brooklyn headquarters, with a number of engineers connected from remote locations via video conferencing. The gathering was for a "Lunch and Learn" session in which the team was expecting to give a presentation on how they conceived of, designed, built, and launched this new personalized feed feature. Six minutes into the presentation, members of the audience noted that the feature was not working.

Members of the team began working on diagnosing and resolving the issue in Etsy's internal #sysops IRC (Internet Relay Chat) channel, asking various questions about the state of the production systems, and began to volunteer diagnostic observations.

A number of engineers moved their virtual discussion into the #warroom channel to focus on diagnosing and resolving the issue. In this #warroom channel, engineers attempted to diagnose what was going on. Engineers discussed various hypotheses about what could be happening. They shared graphs and charts that they noticed with each other, and discussed them. They made suggestions to each other about what actions to take in order to prove or disprove a working theory on the underlying mechanics of the issue.

At 1:18pm, an observation was made that an API call made to populate a subsection of the signed-in homepage called the *sidebar* saw a jump in latency, meaning it was showing much slower performance than normal.

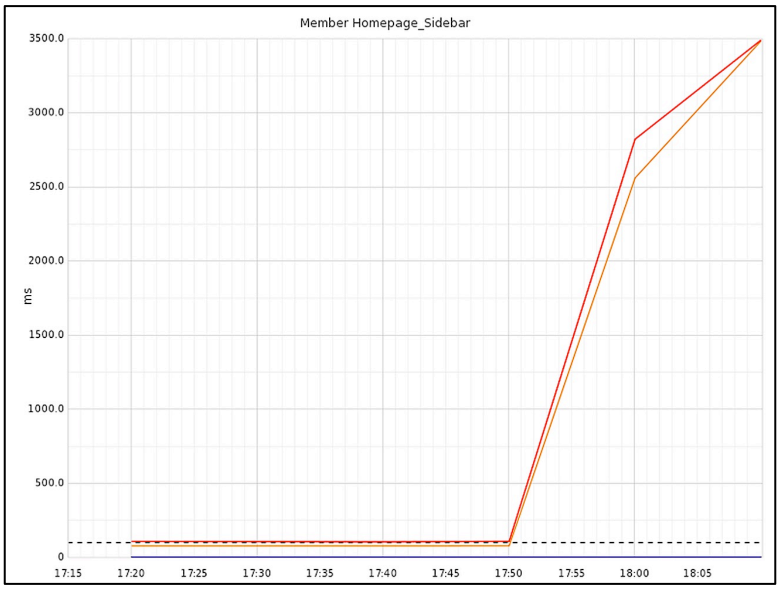


Figure 4 - Median and 90th percentile performance of the Member_Homepage_Sidebar API call

The sidebar section contains a number of links to ancillary content that the user might be interested in and is intended to provide multiple entry points for the rest of the website. This includes pieces of content such as the current featured shop on the blog as well as the links and photos from two most recent blog posts, a post with a featured shop, URL links to top categories, etc. See Figure 5 for a screenshot of the signed-in homepage and locations of the sidebar and its components.

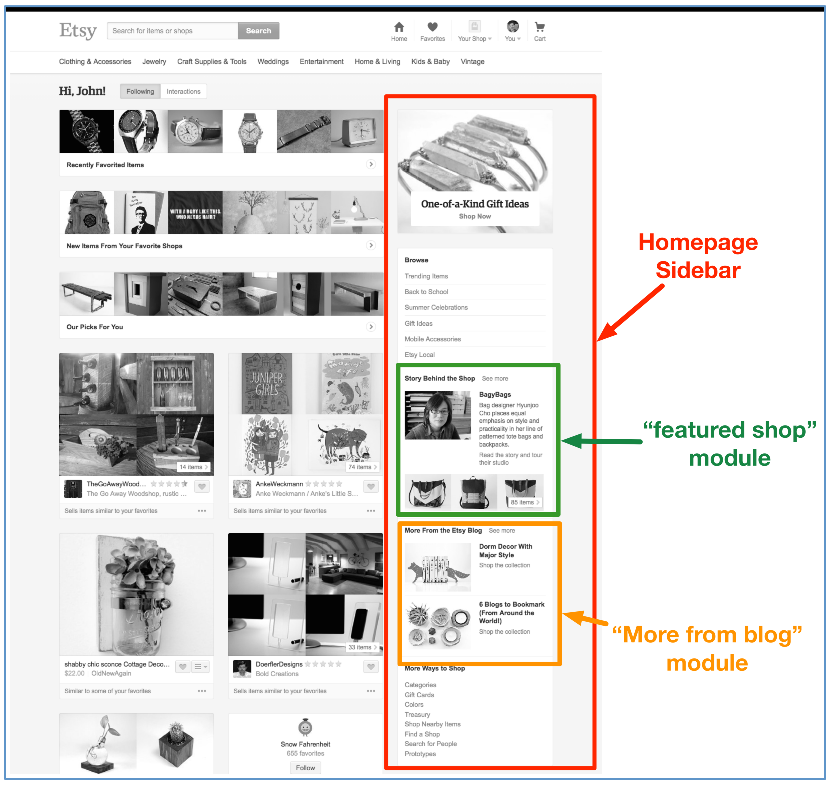


Figure 5 - Signed-in homepage with sidebar components

A suggestion was put to the group to make a change that allowed the homepage to load *without the sidebar*, effectively turning *off* the code that appeared to be experiencing the errors, while leaving the rest of the page intact. This was a configuration feature the team had specifically built for this scenario. Once there was agreement, ProdEng2 began working on making this configuration change.

Members of the team then generated more hypotheses about what could be happening when InfraEng2 posted a line from the application log of a Member_Homepage_Sidebar request:

```
[01:21:01 PM] <InfraEng2>: 10.101.152.3, 10.101.152.3, 10.101.160.45
- - - [04/Dec/2014:18:19:24 +0000] "GET
/v3/public/shops/7887355/cards HTTP/1.1" 400 61
"/v3/member/23636811/homepage/sidebar"
"Api_Client_V3/Member_Homepage_..."
```

which shows a 400 status code response [1](highlighted). InfraEng2 then posted the possible definitions of the 400 status code found in the application code:

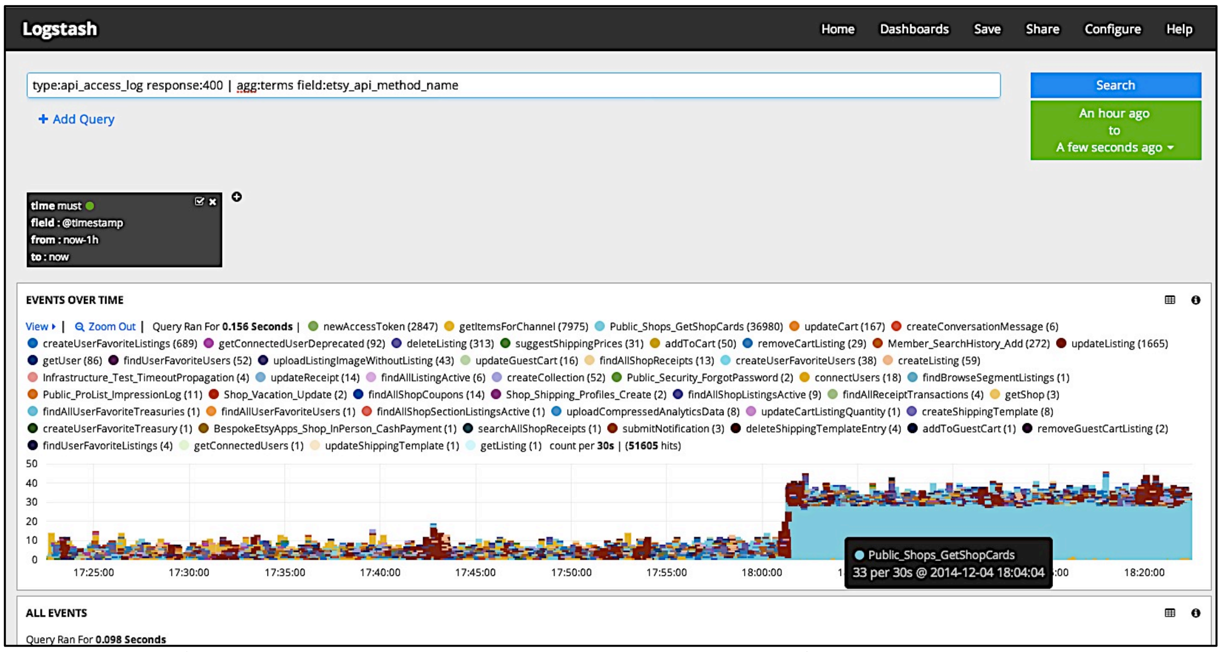
```
[01:21:25 PM] <InfraEng2> 400 == Input_MissingError or
DataType_Exception_InvalidInput or EtsyORM_RecordValidationException
```

InfraEng2 then investigated further examples of the 400 status code.

[1] A 400 status code response in the HTTP protocol signifies a “Bad Request” which is intended to instruct the code making the request to treat it as an error.

The errors associated with the sidebar stopped when ProdEng2 deployed the change to shut off the sidebar, but it remained unclear to the team how the errors were happening when the sidebar was enabled, or how the errors were impacting the personalized homepage.

Observations and hypotheses continued to be shared by the members of the team in an effort to understand the underlying issue. At 1:28pm, an engineer reported that the profile of errors for a specific API method (Public_Shops_GetShopCards) matched the pattern of the sidebar errors. See Figure 6 for the screenshot of the log analysis tool the engineer shared in the #warroom channel.



type:api_access_log response:400 | agg:terms field:etsy_api_method_name

time must
field : @timestamp
from : now-1h
to : now

View ▸ | Zoom Out | Query Ran For **0.156 Seconds** | ● newAccessToken (2847) ● getItemsForChannel (7975) ● Public_Shops_GetShopCards (36980) ● updateCart (167) ● createConversationMessage (6) ● createUserFavoriteListings (689) ● getConnectedUserDeprecated (92) ● deleteListing (313) ● suggestShippingPrices (31) ● addToCart (50) ● removeCartListing (29) ● Member_SearchHistory_Add (272) ● updateListing (1665) ● getUser (86) ● findUserFavoriteUsers (52) ● uploadListingImageWithoutListing (43) ● updateGuestCart (16) ● findAllShopReceipts (13) ● createUserFavoriteUsers (38) ● createListing (59) ● Infrastructure_Test_TimeoutPropagation (4) ● updateReceipt (14) ● findAllListingActive (6) ● createCollection (52) ● Public_Security_ForgotPassword (2) ● connectUsers (18) ● findBrowseSegmentListings (1) ● Public_ProList_ImpressionLog (11) ● Shop_Vacation_Update (2) ● findAllShopCoupons (14) ● Shop_Shipping_Profiles_Create (2) ● findAllShopListingsActive (9) ● findAllReceiptTransactions (4) ● getShop (3) ● findAllUserFavoriteTreasuries (1) ● findAllUserFavoriteUsers (1) ● findAllShopSectionListingsActive (1) ● uploadCompressedAnalyticsData (8) ● updateCartListingQuantity (1) ● createShippingTemplate (8) ● createUserFavoriteTreasury (1) ● BespokeEtsyApps_Shop_InPerson_CashPayment (1) ● searchAllShopReceipts (1) ● submitNotification (3) ● deleteShippingTemplateEntry (4) ● addToGuestCart (1) ● removeGuestCartListing (2) ● findUserFavoriteListings (4) ● getConnectedUsers (1) ● updateShippingTemplate (1) ● getListing (1) count per **30s** | (**51605** hits)

● Public_Shops_GetShopCards
33 per 30s @ 2014-12-04 18:04:04

Query Ran For **0.098 Seconds**

Figure 6 - Screenshot of log analysis shared by engineer, indicating the API method exhibiting the errors

Investigation continued, focused now on the specifics of the API method and its behaviour. It was discovered at 1:32pm that an API request for data on a single shop was producing the errors, and that data for the unique shop ID number did not exist. The user ID of the owner of the shop was found to be an Etsy employee, and that the shop did exist in the past but was in a “closed” state.

At 1:36pm, it was discovered that this shop was linked to from the blog, and the Etsy employee owning the shop had posted an article to the blog about the time that the errors began. An engineer from the content team was contacted, and temporarily “unpublished” the specific blog post. The author of the post was contacted to inform them of the issue as well, and advised not to publish another post until the issue was fully resolved.

The mechanism of the problem was discussed further in the channel amongst many engineers. A hypothesis sequence of this mechanism is suggested:

- A blog post is published, whose author is connected with a shop ID.
- The server code for the signed-in homepage makes an API call to populate the sidebar with data, which includes the shop ID of the blog post author.
- Calls are made to the database for the shop data, which returns a ‘does not exist’ error, and the server returns a 400 status code.

- The 400 status code informs the caching server *not* to cache the empty results, which means every subsequent request for that data will have to generate a query to the database each time.
- Because of the high volume of requests for the shop (on every signed-in homepage request), the caching server begins to queue incoming requests, until the queue gets large enough to incur a 3.5 second latency on the requests.
- The 3.5 second latency is the timeout value set by the homepage code, and it then falls back to a degraded state of generic and non-personalized experience.

With the primary functionality of the signed-in homepage restored (the activity feed) the team decided to leave the "more from the blog" module turned off in the sidebar overnight in order to further validate the causal mechanism and to change the underlying code to gracefully handle 400 status code errors by caching them.

Domain-Specific Background

Architecture Overview

In order to provide the reader a visual representation of the systems involved in the event, participants were asked to draw a diagram of how they pictured mentally the architecture, at a high level. The resulting amalgam of their diagrams can be seen in Figure 7.

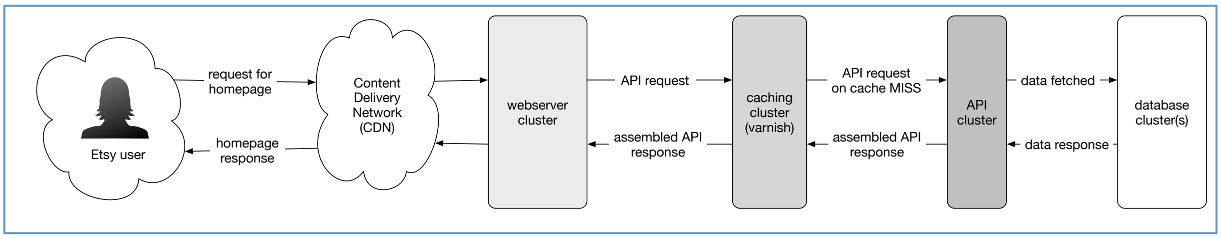


Figure 7 - High-level overview of the systems architecture and typical request-response flow

Typical Request-Response Flow

When a user makes a request for the logged-in homepage of Etsy.com, it passes through a Content Delivery Network (CDN) and is received by a cluster of webservers that then execute any number of API requests to a cluster of caching servers running the varnish software. If the exact request results are found in the cache (a cache "HIT"), then the cache server will return the data directly. This is intended to provide a performance and efficiency benefit. If the request results are not found in the cache, it will forward the request on to a cluster of API servers, which will then retrieve the necessary data from various database clusters that store the data. These results will then be cached as the response passes back through the caching cluster to the webserver, where the data is formatted into a response the user's web browser can interpret.

In the case of this event narrative, the API request method (*Member_Homepage_Sidebar*) made by the webserver to the API in turn executed a call to the *Public_Shops_GetShopCards* request method for the author of the most blog post in the "More from the blog" module, which was found to be closed. The API method interpreted this as "does not exist" and returned a 400 status code, which brought the caching server(s) to not cache that result. (Typically, caching servers will choose not to cache "negative" responses by default, based on the assumption that it's better to continue going back to the canonical data source each time rather than serve a cached error response even after the canonical data is corrected.)

Since the results are not cached, it adds latency to the response time; every new request will result in the entire process repeated. This response time increase then keeps processes in the caching server busy for a longer period of time, and the behaviour of the caching software is to queue requests that haven't been satisfied yet due to a limited number of processes available. With many users logged in during the holidays, more and more requests for non-existent shop information arrive and eventually the requests pile up enough for the feature to be considered unusable.

Caveat of constructed narratives

A caveat must be made for the narrative offered above. What is presented here is not intended to provide a comprehensive and objective account of the event, as that would be impossible. Instead, the purpose of the narrative is to offer a high-level timeline of selected details and junctures within the event in order to situate the study's focus, and provide a scaffold upon which the analysis and approaches can sit.

RESULTS

This research focused on eight of the twenty engineers that responded to the degradation, as they were the majority of responders in the #warroom channel during diagnosis and response activities found in the transcript. The remaining 12 engineers' questions and reflections appeared in the transcripts, but their utterances were not seen to significantly influence or represent meaningful data in the study.

Coding Schema: Listening To The World

An initial coding of the IRC transcripts was done in order to categorize the utterances into categories. This "first cycle" coding (Saldaña, 2009) was exploratory in nature, and the resulting codes can be seen in Appendix A.

The initial coding schema provided focal points to use in the pilot interview, which was with InfraEng1. After a transcription was made of the pilot interview, the coding schema was then transformed into a tighter focus around critical junctures and episodes of uncertainty. This schema was then used to prepare questions for the remaining interviews with participants in order to explore what tacit knowledge or heuristics could be gleaned from the cued recall of events.

Timeline Overview and Map

Episodes of activity were constructed from the transcription data and validated via the interviews. The episodes identified fell into three categories:

- *Coordinative activities*, which includes multiple engineers supporting and sequencing tasks in order to produce an outcome,
- *diagnostic activity* which includes hypothesis generation and relayed observations, and
- *disturbance management* activities, where the state of the response to the event was discussed or requested.

The identification of heuristics focused largely on the diagnostic activities and coordination activities found in the data. These focus areas were selected for two reasons:

a. The necessity of reducing the available volume of data, to contain the scope of the research.
b. Initial coding indicated both areas involved participants coping with uncertainty. Focusing on these areas increased the likelihood of identifying any heuristics in play.

A timeline view of the utterances made by participants in the #warroom IRC channel during the event can be seen in Figure 8.

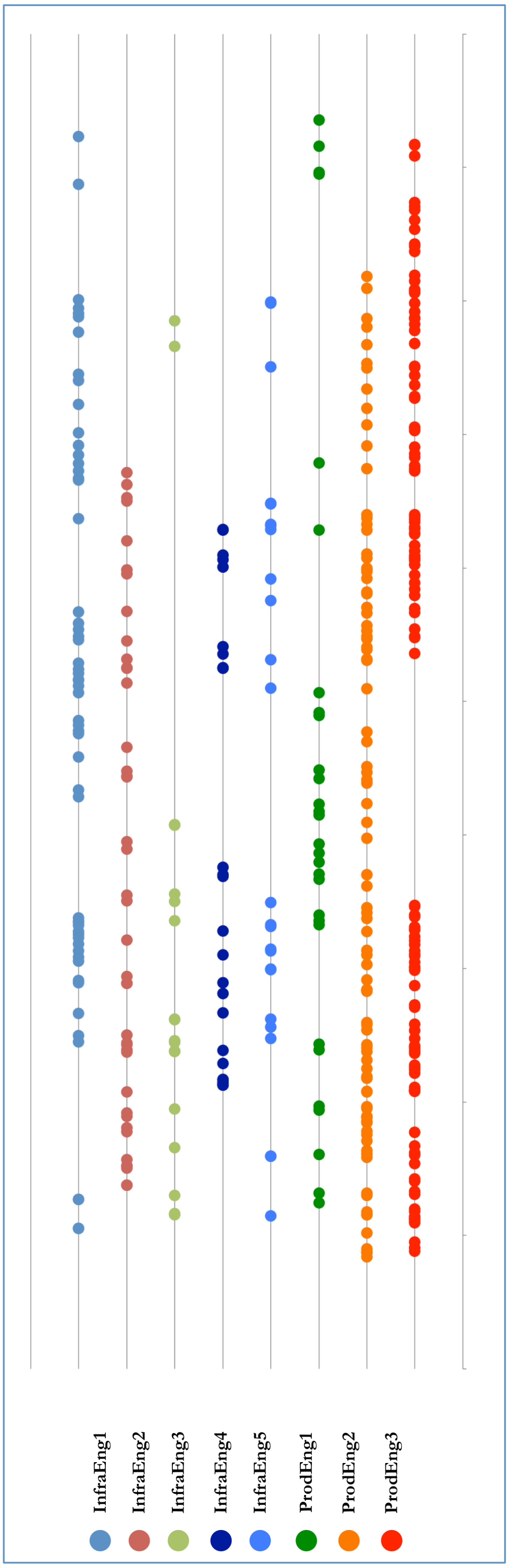


Figure 8 - Timeline view of utterances in IRC, by participant

The non-verbal behavioural data were superimposed on the overview timeline of the event. For individual participants, this included both the coded utterances from IRC transcripts with "diagnostic activity" and "taking action" top-level codes. The behavioural timeline data included, in some cases:

a. Dashboard accesses
b. Staff directory accesses
c. Requests to the Princess environment as changes were being made
d. Requests to production site (https://www.etsy.com)

The individual traces can be seen in Appendix C.

In addition to the requests participants made to the production site during the event, data was collected on each participant's access specifically to the signed-in homepage during the event. See Figure 9.

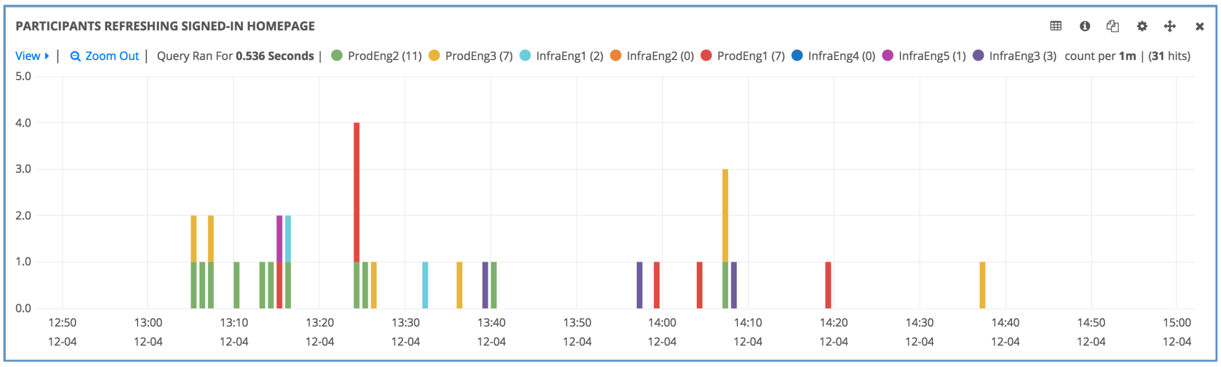


Figure 9 - Participants refreshing signed-in homepage

Coordination episodes

There were five episodes identified as coordinative activities in the event, each of them aimed at making changes to production systems in order to either limit the exposure of the degradation mechanism, or to restore normal functionality.

A timeline map of all utterances grouped by diagnostic activity and action taking/response can be seen in Figure 10, with the five coordinative episodes indicated.

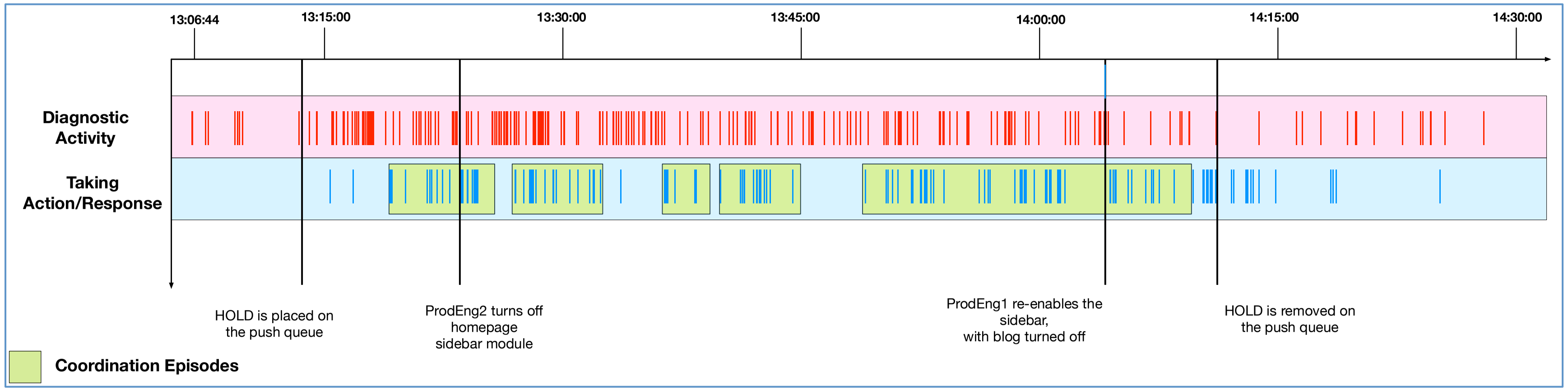


Figure 10 - Timeline of event with diagnostics/response/coordination episodes

Episode 1

Beginning at 1:19:10pm, in response to InfraEng1 and InfraEng2's discovery that the homepage sidebar functionality was logging errors somehow implicated in the behaviour they were observing, ProdEng2 suggested that the sidebar on the homepage should be "turned off" or otherwise skipped in the execution of the homepage code.

This episode in the #warroom transcript was coded with three different codes:

- Suggestion of action/request for feedback
    - Example: "should I shut that off?" (referring to the sidebar portion of the homepage)
- Feedback on an action/artefact
    - Example: "lgtm" (looks good to me) (referring to the proposed configuration change to deploy)
- Asserting the state of an action being taken
    - Example: "going to prod" (referring to the act of deploying the change to production systems)

The coded details of the entirety of coordination episode 1 can be found in Appendix B. A visual representation of the episode on the event timeline with the above codings can be found in Figure 11.

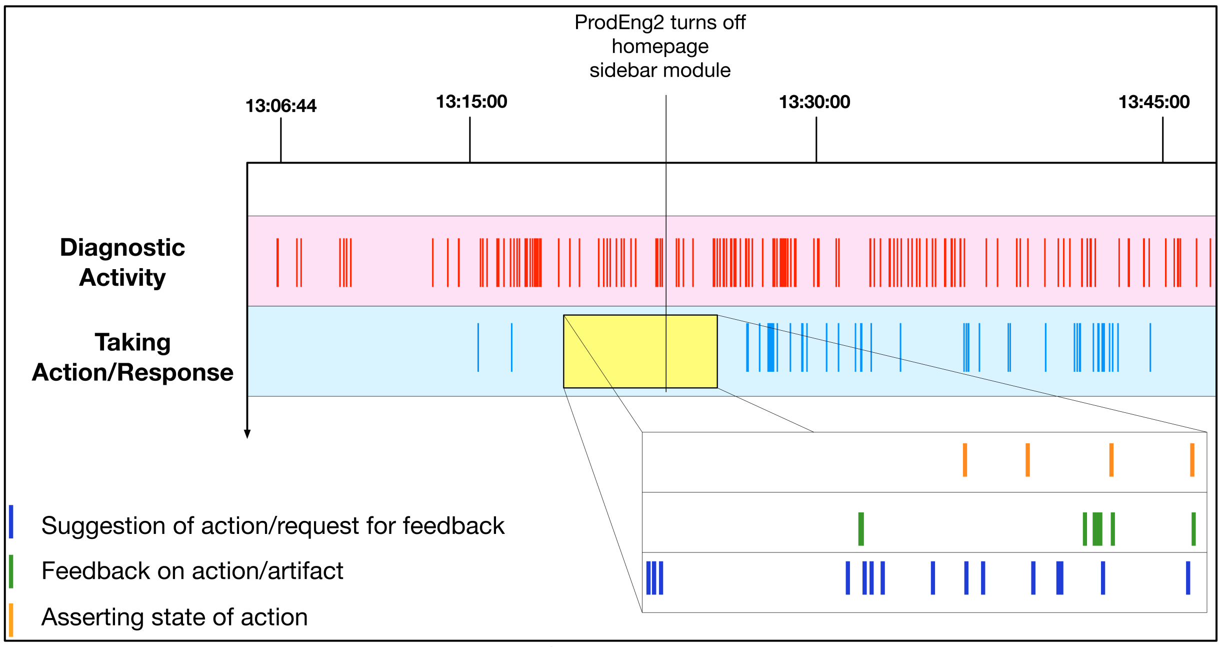


Figure 11 - Coordination Episode 1

Episode 2

At 1:24:44, ProdEng1 hypothesized: "something wrong with the featured shop?" The featured shop is another module on the homepage sidebar, shown in Figure 5. Specifically, the module is a blog post whose content is the profile of an Etsy shop owner highlighted by the editorial staff at the company. ProdEng2 suggested re-enabling the sidebar, but turning off the featured shop portion of the sidebar, and asked for feedback on the idea.

Asked about this suggestion, ProdEng2 suggested that turning off the featured shop module was a fruitful way of implicating it as part of the issue:

> *"I think at this point, we had gotten ... <InfraEng2> had uncovered a really, really good clue. We had successfully verified it was the sidebar. Not just in theory, not just from evidence, but we shut the sidebar off and the errors go down. Now I was in, and I think ProdEng3 was also in agreement with the mode of, "Let's get hardcore evidence that it's the featured shop." We know it's something in the sidebar, let's turn the sidebar back on, sans the featured shop and see if these error come back, right?" That's the easiest way for us to verify whether or not it's the featured shop."*

After ProdEng2 asked for feedback on the configuration change, InfraEng1 asked:

```
[01:30:31 PM] <InfraEng1> <ProdEng3>, <ProdEng2>, can we hold on
that?"
```

at which point ProdEng2 acknowledges the request to hold off on not deploying the change at the moment, and the episode surrounding making the change is temporarily abandoned.

A visual representation of the episode on the event timeline with the above codings can be found in Figure 12.

Between episodes 2 and 3

At 1:32:24, InfraEng2 relays an observation in IRC:

```
[01:32:24 PM] <InfraEng2> here's the shop error:

[01:32:25 PM] <InfraEng2>
InfraEng2@InfraEng2:~/development/Etsyweb$
curl http://server:port/v3/public/shops/12345678/cards

[01:32:25 PM] <InfraEng2> {"error": "Shop 12345678 does not
exist"}
```

InfraEng2 then asks:

```
[01:32:51 PM] <InfraEng2> so why are we loading that shop?
```

Both ProdEng3 and InfraEng5 reply that they've looked up the shop's details and discovered that it is owned by an Etsy employee, and links to the employee's staff page on the company's internal directory. ProdEng3 also notes that the shop is marked as currently closed.

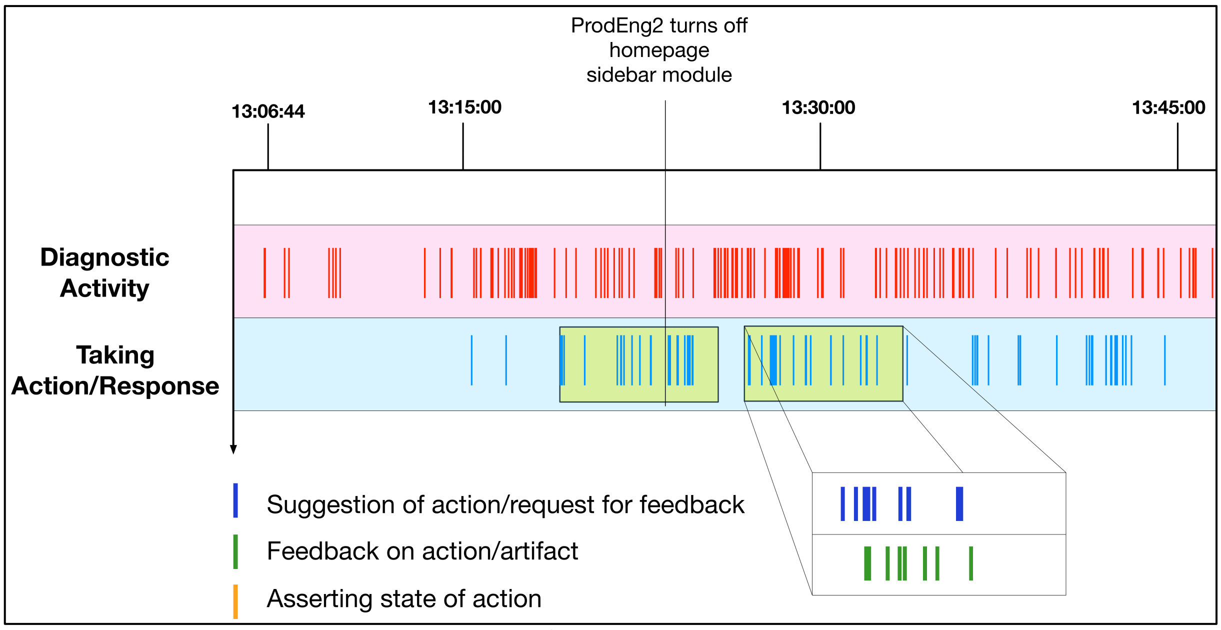


Figure 12 - Coordination Episode 2

Episode 3

At 1:36:04, ProdEng1 makes the observation that the shop found by InfraEng2 that is returning errors for being closed is currently on the blog's most recent post which would appear in the "more from the blog" module, not the "featured shop" post and posts a link to the blog post in question.

ProdEng2 then suggests again re-enabling the sidebar, but turning off the featured shop. ProdEng3 then replies that it is necessary is to turn off the "more from the blog" module, not the "featured shop" module. ProdEng2 acknowledges the suggestion, and asks for feedback on a configuration change to proceed with that plan of action.

A visual representation of episode 3 can be seen in Figure 13.

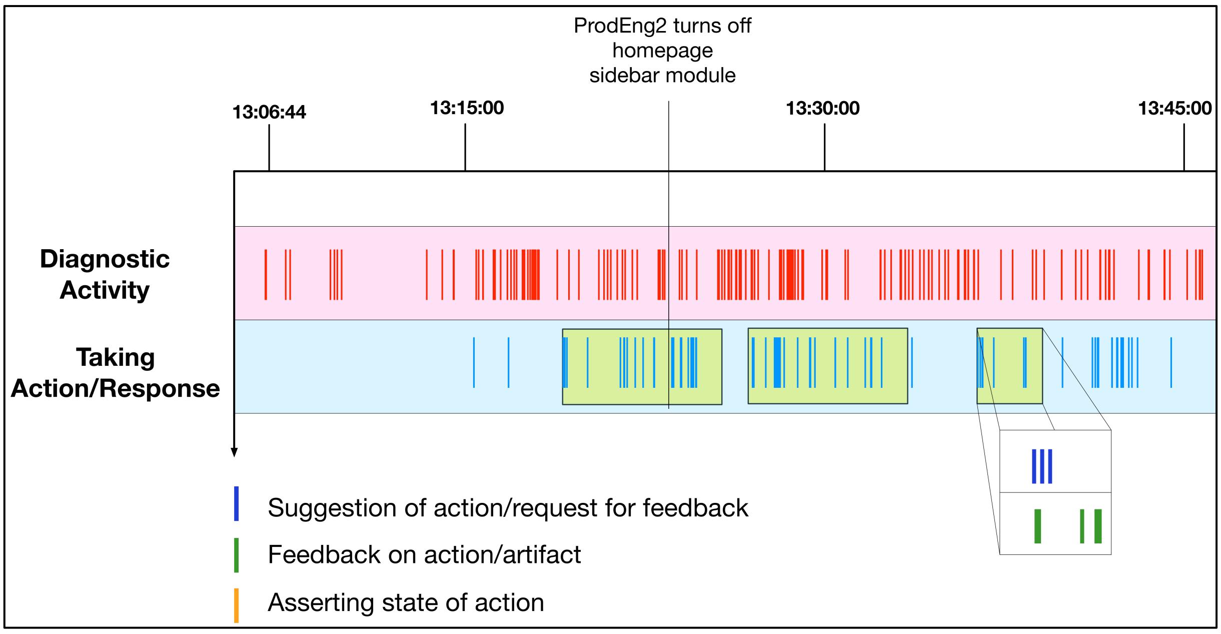


Figure 13 - Coordination Episode 3

There is no feedback from the team on ProdEng2's suggested change, but instead dialogue about contacting the editorial team and the author of the blog post. InfraEng4 states:

```
[01:39:04 PM] <InfraEng4> i feel like we're still unclear on
what we're fixing

[01:39:12 PM] <InfraEng4> the blog or the featured shop?
```

The intended course of action (turning the sidebar back on, but turning off the "more from the blog" module) does not proceed further.

Episode 4

After the blog post was identified, a member of the team that manages the blog infrastructure is requested to join #warroom. They offer to unpublish the specific blog post and there is agreement from members of the responding team.

There is then confirmation that the post was indeed unpublished and any cached versions of the post had been purged. A visual representation of episode 4 can be seen in Figure 14.

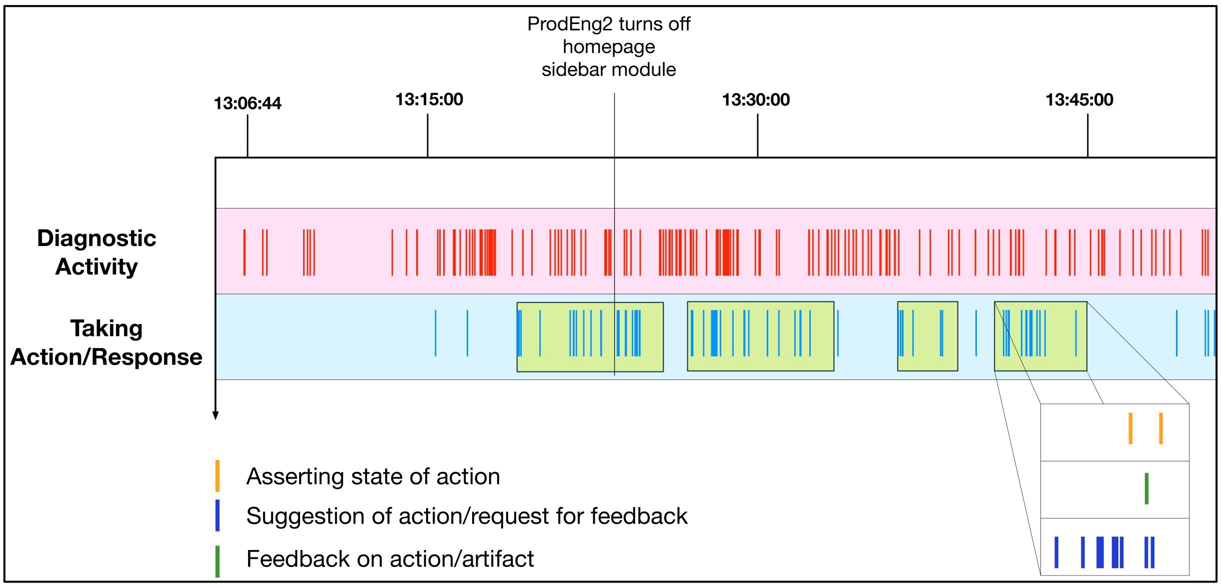


Figure 14 - Coordination Episode 4

Episode 5

After the implicated blog post had been unpublished, the suggestion is again raised at 1:49:07 by ProdEng1 to re-enable the sidebar and turn off the "more from the blog" module. As the configuration change is prepared by ProdEng2, engineers continue hypothesizing about how a closed shop would produce the effect they were observing.

A visual representation of episode 5 can be seen in Figure 15.

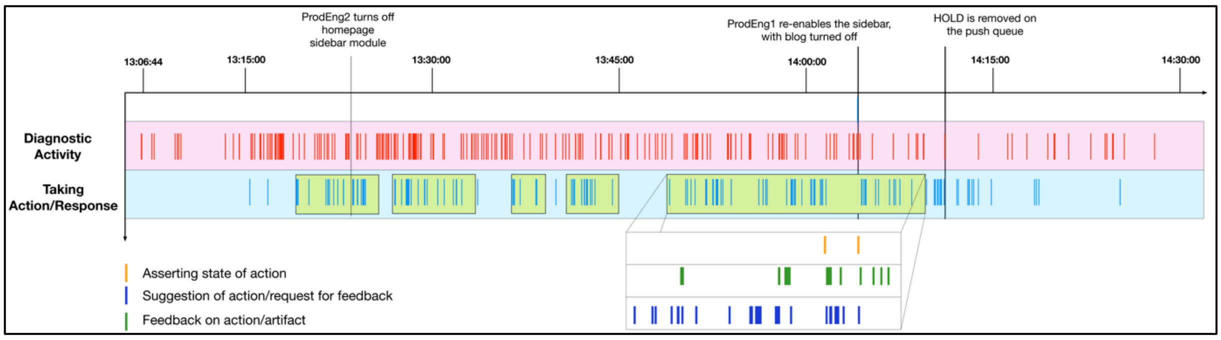


Figure 15 - Coordination Episode 5

Diagnostics and Hypothesis Generation

As the event unfolded, both the #sysops and #warroom transcript indicated that engineers were generating hypotheses, sharing them with others, and making efforts to validate or test them.

Hypothesis generation in the diagnosis of the issue was found to manifest in different ways, and this is reflected in how the IRC transcripts were coded. The first and most straightforward way is simply a stated hypothesis, such as:

"I wonder if Wordpress is dying?"

A second manifestation of hypothesis generation is indicated by questions directed at a specific topic. An example found in this case is:

> *"Did we just turn off a CDN or make F5 changes?"*

This can be seen as a search for data, used to construct a candidate hypothesis. Thirdly, the relaying of observations to other members of the team was also categorized as a strong inference of hypothesis generation. For example:

> *"InfraEng2: this shows the same problem with GetShopCards <URL link to log search>"*

Both relaying of observations and questions or statements of hypothesis were coded as diagnostic activity. Examples of how diagnostic activity was identified and coded can be found in Appendix A.

A visual representation of hypothesis generation and key observations can be seen in Figure 16.

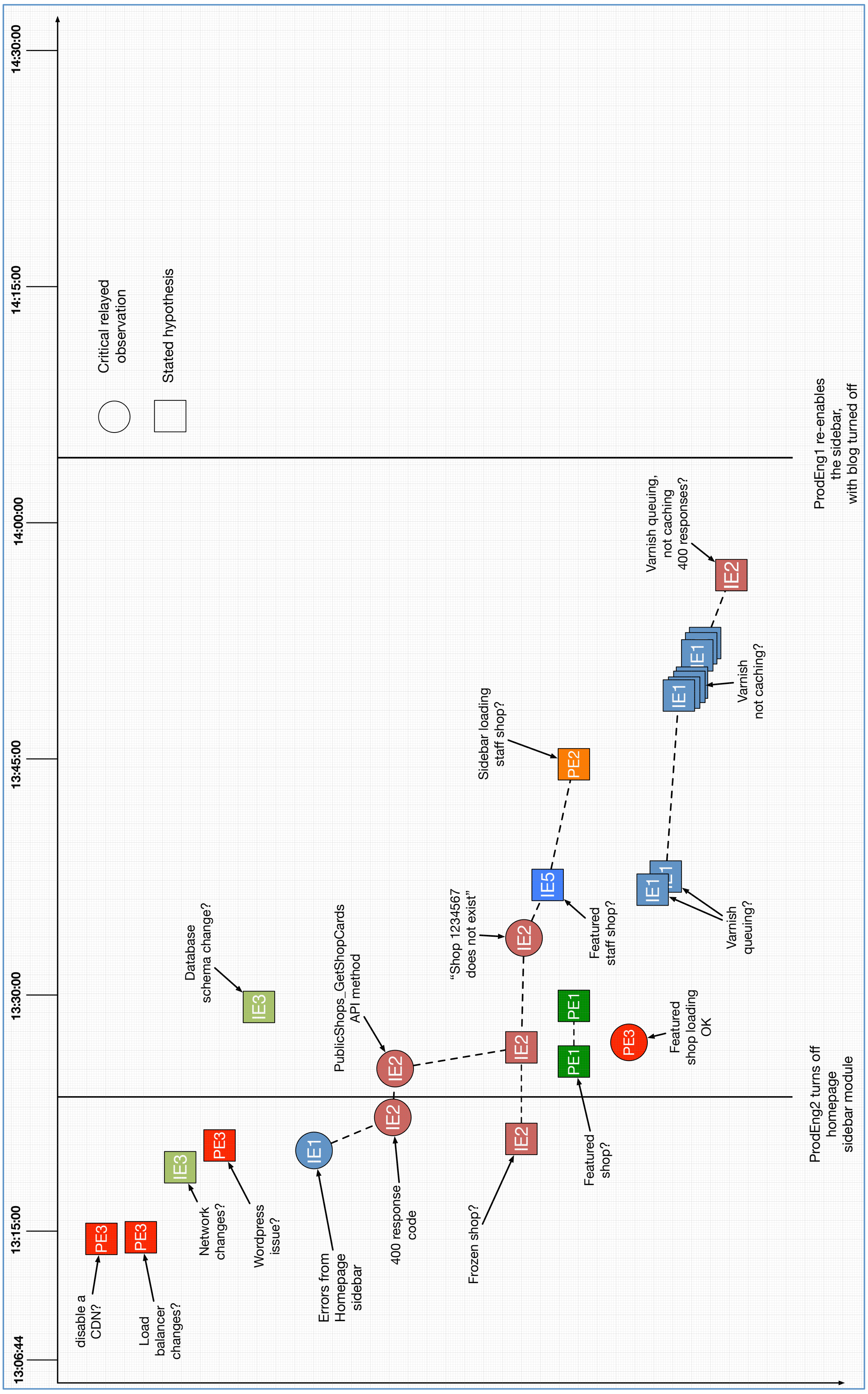


Figure 16 - Diagnostic map of hypotheses

Hypothesis Generation and Origins

As indicated in the Figure 16, a number of hypotheses were generated and shared in the IRC transcript as the event unfolded. Some of the hypotheses were stated and then invalidated in some way, while others proved to be fruitful directions that brought other engineers to explore.

Hypothesis #1: CDN or load-balancer changes

At 1:13:28, ProdEng3 asked in the #sysops channel:

```
[01:13:28 PM] <ProdEng3> did we just turn off a CDN?
[01:13:34 PM] <ProdEng3> or make f5 changes?
```

And the answer received is no, there have been no changes to the F5 (a server load-balancing device), or a CDN having been just turned off. At 1:15:20, InfraEng2 suggests that engineers responding to the outage join the #warroom channel, and then InfraEng5 also requests information about a CDN's on/off status:

```
[01:17:54 PM] <InfraEng5> <CDN vendor> has been off for a while
now, right? that didn't just happen?
```

When asked about what brought them to ask for that information, both ProdEng3 and InfraEng5 remarked that the initial signals of the outage reminded them of a similar previous outage.

> *"This matched a pattern that we saw earlier. By earlier, I mean many months earlier. When we turn off the CDN that was doing response buffering. I'm sorry, it was doing post-buffering."*
>
> *"We were seeing timeouts on that particular endpoint and also just again the fact that we were seeing idle workers plummet and busy workers skyrocket. Pointed me towards this pattern that we had seen before." (ProdEng3)*
>
> *"We've seen patterns in the past where we turned off one CDN that was fronting the API and it changes the profile of the request rate to the API… so if we see a increase in request for second to the API cluster can be an indication that we turned off the CDN." (InfraEng2)*
>
> *"There was a prior event related to API errors, specifically with <vendor>, and I think it was within some fairly recent time ago. It may have been ... It's at least on the deploy dashboard, a graph showing the balance of CDNs, and <vendor>t at this point was off. I think I was asking if that change had just occurred, if we had just altered the balance of CDNs, because I knew from some prior event that we had some poor interaction with <vendor> because we were debugging API errors. It must have been something I saw. I don't remember it that well." (InfraEng5)*

As for the F5 load balancer diagnostic direction, ProdEng3 described that they had seen recent mention of work being done on the network:

> *"The outcome of the previous instant that I was just discussing with the apps and CDN was that we turned on buffering on the F5s. I had seen some chatter about some NetOps work being done. I was curious if maybe we change that configuration to disable buffering there, which would have put us in the same state. If we were relying on buffering occurring on, at the F5s, it would have put us in the same state that we had in the prior instant." (ProdEng3)*

There was universal agreement among other participants when asked whether CDN status was a reasonable diagnostic direction to take.

Hypothesis #2: Network changes

When asked what brought them to state this in #warroom:

```
[01:18:01 PM] < InfraEng3 >: and #netops says no network changes
```

They remarked that they had seen the previous questions about CDNs and load-balancers, and considered network changes:

> *"Probably what happened is I saw ProdEng3 speculating and I went to go find answers."*
>
> *"Yeah, yeah. So the first two questions were about CDNs, and I found the answer to that. So when InfraEng5 brought it up later, I hadn't had that answer. And the second one either I didn't at that point in time realize how easy it was to see network deploys, or I wanted to verify that something about how we graph network deploy would've been wrong." (InfraEng3)*

Hypothesis #3: Externally-hosted blog

Blog content at Etsy is hosted on a third-party Wordpress host, and the homepage contains two main blog-linked modules: the "featured shop" and "more from the blog" modules. Both of which link to the blog, which is hosted outside of Etsy's infrastructure.

At 01:19:24, ProdEng3 asks:

```
[01:19:24 PM] < ProdEng3>: I wonder if wordpress is dying
```

When asked about the origin of this hypothesis, they remarked:

> *"The blog is, it's an endpoint that had caused us trouble in the past because it has been slow in the past. The sidebar prominently has blog posts in it."*

Other participants also believed that Wordpress was a reasonable diagnostic direction, explaining that it 1) has had issues before, and 2) lives outside the control and visibility of Etsy's staff.

> *"I think at this point, if I recall, we were all having a hard time finding graphs that were moving the way we wanted them to, to validate assumptions. If WordPress was having issues, we can't graph it, it's outside of our control. I think ProdEng3 may have been thinking outside of the Etsy infrastructure, a little bit." (ProdEng2)*
>
> *"He probably thinks to himself: well what are the components of the sidebar? where do they call out to? And one of them calls out to something external, so it's a red flag." (InfraEng3)*
>
> *"Historically we've had problems with the blog not loading and causing problems like this. My hypothesis would have been that it was the blog. When I say the blog, it's the blog runs on different infrastructure than Etsy." (ProdEng1)*
>
> *"If I remember, it's hosted by someone else. That's a big part of that hypothesis. Is that, and it's also Etsy's code, it's WordPress. When I see the sidebar is failing, it's like: 'oh it's probably*

WordPress problem or <blog host vendor> problem' or something or whatever. I don't know if that's even what it is." (ProdEng1)

Hypothesis #4: Frozen Shop

At 01:19:49, InfraEng2 asks the #warroom channel:

```
[01:19:49 PM] < InfraEng2 >: did some shops get frozen?
```

A shop is "frozen" (suspended) by Etsy's Member Support team in particular scenarios, such as a late payment on a bill, or a suspected fraud activity. It is intended to be a temporary state for a shop until the status is cleared up via Member Support's communications with the seller.

When asked what brought them to ask this in the channel, InfraEng2 stated:

> *"Something that is not specific to the API that we've seen before is that we see this sudden increase in errors where if Trust and Safety closed a the shop or a series of shops that have a lot of listings that have external links to them, we suddenly see a spike of errors where incoming traffic was not from people searching for listings on the site, but clicking on links from on other sites to us, hit a listing page that is no longer available and we don't show a message that says the listing is no longer available but we show a either a three-armed sweater or a whoopsie page. So we see this sudden spike in errors that are unrelated to any of our changes that we've made in terms of pushes or code deploys but Trust and Safety has just made a data change, which then means that the app behaves differently."*

No other engineers in the transcript arrived at this hypothesis, and while many reported in the interviews as being able to see how InfraEng2 came to state this, they did not recall understanding the diagnostic direction at the time.

Hypothesis #5: Database schema changes

At 01:28:12, InfraEng3 asked for information along a diagnostic direction involving databases:

```
[01:28:12 PM] < InfraEng3>: are we doing a schema change right
now?
```

Database schema changes are modifications to production database structures. Typically, schema changes pose no issue to production services, as there are two mirrored databases for all data, replicating the data between themselves. A schema change is made to one of the pair while it is removed from production, and then it's returned to production in order for the remaining server to be modified.

When asked about what brought them to ask about that, InfraEng3 replied:

> *"You know that feeling when you have seen a problem recently, so it's like primed in your head as a possible problem? This was because maybe like a weekend or two before this was the weekend we had the weird materials pinning problem on <database> that we later discovered was a problem with like all of <database>."*

"And I think my weirdo theory here was something like if there was somehow replication broke for the featured shop, and it only existed on the side that was out for schema change and we featured it. "

Disturbance management and tracking the state of the response

Indications of disturbance management activity were found in the transcripts and in interviews. This type of activity differs from diagnostic activity or developing plans of action in the response. (Woods, 1995a) describes disturbance management as referring to the "interaction between situation assessment and response management (response selection, adaptive planning and plan generation) that goes on in fault management in dynamic process applications." (p.74)

Examples of this:

```
[01:30:33 PM] < InfraEng4> is there a rush to get this deployed
today? should we slow down?
[01:32:27 PM] < InfraEng4>: what is our current state?
[01:34:10 PM] < InfraEng4>: wasn't the sidebar not loading
before? trying to understand what the toggle of the flag did
[01:39:04 PM] < InfraEng4>: i feel like we're still unclear on
what we're fixing
[01:39:12 PM] < InfraEng4>: the blog or the featured shop?
[01:53:26 PM] < InfraEng4>: have we communicated to the employee
in question?
[01:58:24 PM] < InfraEng4>: ProdEng2: if there are still this
many questions, i'd prefer we skip the blog post for now until
we know why it's killing us
```

Concern about the pace of the response and diagnosis appeared to be a significant driver of this status-requesting and response management activity. InfraEng4 reported:

"In general, I think the most difficult thing is the stress that is involved in diagnosing and it gets in your way, I think. I think a lot of my line of questioning was along the theme that slow is fast in these situations. We don't want to be reacting too quickly without understanding where we at. We don't want to be jerking back and forth between up and down, and up and down. It's not a time to really shotgun a solution in just terms of, 'let's try everything and see if it works.'

"I think the biggest part is slowing down and buying us enough time so that we can get deeper into the problem, and actually figure it out. Had <engineer> pushed their solution too early, we might have not gotten any insight. What would the game been there?"

ANALYSIS

Levels of participation

A high-level observation was that ProdEng2 and ProdEng3 comprised roughly 52% of the dialogue in IRC, with the remainder made by the other participants. See Figure 8. ProdEng1-3 were most familiar with the functionality (the signed-in homepage) in focus, as well as the team which designed the controls for graceful degradation of the functionality, so their level of involvement is not surprising.

Findings in Coordination Episodes

Themes that emerged from analysing the coordination episodes:

a. **Graceful degradation controls as response.** Almost all of the activities were aimed at *reducing* the untoward effects of the degradation and stabilization of the site's functionality, as opposed to immediately focusing on full recovery.

   - Activities in episode 1 were focused on bringing the signed-in homepage experience back from the fully degraded state (the generic "trending items" display, which is the failsafe when API calls take longer than 3.5 seconds) to a less-degraded state (returning all personalized content for the user such as the activity feed, but without the sidebar content).

   - Activities in episode 2 were focused on bringing the homepage to a less-degraded state by re-enabling the sidebar without the "featured shop" module. This effort was abandoned upon the discovery that the shop referenced in the "featured shop" was not problematic.

   - Activities in episode 3 were focused on bringing the homepage to a less-degraded state by re-enabling the sidebar without the "more on the blog" module, and was stalled or abandoned.

   - Activities in episode 4 were focused on unpublishing the blog post that was identified as problematic due to referencing of the shop owned by the author.

   - Activities in episode 5 were focused on bringing the homepage to a less-degraded state of sidebar turned off to sidebar turned on but without the "more on the blog" module.

   This focus on moving between more to less degraded states was reflected explicitly in the dialogue during episode 2:

```
[01:31:46 PM] <InfraEng4> i guess i'm asking if we need to do
root cause determination right this very instant

[01:32:16 PM] <ProdEng3> InfraEng4: not really but it'd be good
to be in a less-degraded state if possible
```

   The interviews support this perspective as well:

"When we worked on the homepage degradation stuff, we defined a series of degraded states that are kinda cascading. At the bottom, we were showing the trending items and only the trending items. We are, as degraded as a homepage can get, without showing an error page. My instinct in a situation where the site's down, there's an error page, there's a degradation and something like that is, to elevate up as high up, to a less degraded state, as quickly as possible. Then retroactively, try to identify why we degraded it in the first place. My instinct was, the theory is that the sidebar's causing all the problems. If we can shut the sidebar off, we can bump up to, not a very degraded state, to a state that's almost the full homepage. Then, we can go back and start spelunking logs, looking at graphs." (ProdEng2)

"From a product perspective, I think that it's best to have the software, the product that you're providing to a user or a customer, be in the most functional state that you can get it in. To me, the priority is over ... One way that I'm thinking about your question is, we could leave it in the state that it's in, and then find out what the root cause, and then fix that. Then we go from totally degraded to not degraded at all. Versus my instinct of, "Let's bump off the degraded states a few levels, and then continue our search." (ProdEng2)

"To me, with an unknown amount of time until we figure out what the problem is, and it still seemed like we had no idea. I think at that time, we still had relatively little idea, what was causing the sidebar to fail. It was like, "Well, let's get up a few levels. The users can actually use the homepage again. It's kind of close to the state that it's designed to be used in. It's not optimal, but it's better than where we are. Then, we can continue to diagnose, and then hopefully we can bump up back up to the full thing, instead of waiting." (ProdEng2)

When asked about his reply to the suggestion made to move to a less-degraded (but still degraded) state in episode 2 ("it's worth a try"), ProdEng3 remarked:

We are showing the actual…the bulk of the homepage at this point. However, we are still missing out on the sidebar. This was ...the sidebar, it's important. It introduced people to... it introduces people to our browse categories and it gets people diving further into the site.

This was…we were in the Holiday season and my goal at this point was to have our product as close to 100% as possible." (ProdEng3)

b. **Peer Support for making changes.** When suggestions are made to change live production systems, engineers relied almost exclusively on peer consensus on whether to take action or not, and discuss any details surrounding the change. In the case of episodes 2, there was explicit feedback to *not* move forward with the change, and in episode 3, there was an absence of feedback, and the suggested change did not move forward.
c. **Not waiting for automated tests to finish before deploying to production.** Twice during the episodes, suggestions were made to the engineer to not wait for the automated tests to finish running.

Identified heuristic usage and anomaly response patterns

The fundamental finding in this work is that engineers do employ context-specific heuristics in outage scenarios, as opposed to a singular reliance on prescribed responses or procedures.

There were four heuristics identified in this case. The first three are centred on diagnosis:

1. Heuristic #1: in a tacit effort to narrow the scope of a diagnostic search, first look for any correlation to the last change made to the system(s).
2. Heuristic #2: if no correlated change is implicated as contributing to the system's aberrant behaviour, then widen the diagnostic search space to any potential signals
3. Heuristic #3: if no correlated change is implicated as contributing to the system's aberrant behaviour, then there are two primary diagnostic directions to take:
   a. Look to confirm/disqualify a specific and past diagnosis that comes to mind by matching signals or symptoms that appear similar.
   b. Look to confirm/disqualify a general and *recent* diagnosis that comes to mind by matching signals or symptoms that appear similar.

And the fourth is centred on coordination:

4. Heuristic #4: in outage scenarios, use peer review of any code changes to gain confidence, as opposed to other methods, such as automated tests or other procedures.

Heuristic #1 – What did we just change?

Perhaps one of the most surprising findings of this thesis is that the most prevalent rule-of-thumb discovered was one that was *least* explicit in the IRC channel dialogue. All of the participants reported that upon becoming aware of an outage or degradation happening (whether through alerting, observing the behaviour first-hand, or being informed by others) before gathering *any* diagnostic data, their first instinct is to look to see if a change has been made to the application. In almost every case, this manifests in answering a question:

"Does this line-up with a deploy?"

The only two mentions in the IRC transcript about correlation with code deploys were:

Approximately 12 minutes into the outage:

```
[01:17:28 PM] < ProdEng3 > doesnt line up with any push
```

and again, approximately 30 minutes into the outage:

```
[01:35:40 PM] < InfraEng4 > ok. so this was not deploy related?
[01:35:44 PM] < ProdEng3 > it doesnt seem to be, no
```

Participants expounded on the importance of looking for this correlation in interviews.

"Honestly, that's probably the first or second thing I do if anything goes wrong." (InfraEng3)

"I'd say it's high (on the list of things to check). It's probably almost number one." (ProdEng1)

"It's something I reach for, pretty quick, I think when I see a graph spike, to get the push lines drawn on it. Yeah, I feel like change is probably one of the biggest causes of an outage or an error, in software engineering." (ProdEng2)

"It's just like…usually the thing is fine, and then someone changes something, then it breaks. You just want to see what the most recent change is." (ProdEng1)

> *"The easiest thing is: someone did a code push and it altered something that we need to fix." (InfraEng2)*

When asked about the rationale for first seeking what changes might have been made recently (as opposed to any other behaviour) participants suggested some fundamental reasoning.

> *"Two reasons. One is because I feel more comfortable with code changes than I do with – and like looking. And I think we have good visibility into those. Like you go pull out the deploy dashboard. You look at if the lines coincide. You can dig into the commits that are around that time. You can see if any of them match that part of the codebase. Like it's very – it feels really tractable." (InfraEng3)*
>
> *"…I actually feel like systems at rest don't break." (InfraEng3)*
>
> *"If it had lined up with a push, then that pretty clearly indicates that someone had just changed some code somewhere. If that's the case, we have a handful of commits that were in the push, that we could start looking for candidates that could have broken something. I think he was probably telling people, that maybe that was not a route to pursue, we should look for other things that could have caused the outage." (ProdEng2)*
>
> *"It would probably be one of the first things to look for, but the main reason why I say that is pretty specific to ... actually you know what? Even thinking about my experiences outside of Etsy, it's almost always one of the first. Even in places where there's not good graphing, good metrics gathering or anything like that, good monitoring. When something breaks, one of the first questions I think I've asked is, "Did anyone just push or did someone deploy something? Did someone change something, to cause a breakage?" (ProdEng2)*
>
> *"Those represent the most volatile kind of change that we do regularly, in terms of amount of changes is an amount of entropy introduced. Those are the most frequent and most visible." (InfraEng5)*
>
> *"We assume that our natural state is operational. When we move out of that state, my assumption would be that something has occurred that introduced a different behaviour. That's why I'm looking at it." (InfraEng4)*

A number of participants alluded to the use of a specific feature of dashboard design at Etsy (internally referred to as 'tooling') that supports this heuristic already, which are "deploy lines" – vertical lines that are drawn on time-series graphs when a code change is made, to allow correlation to be determined visually. See Figure 17 for a typical example.

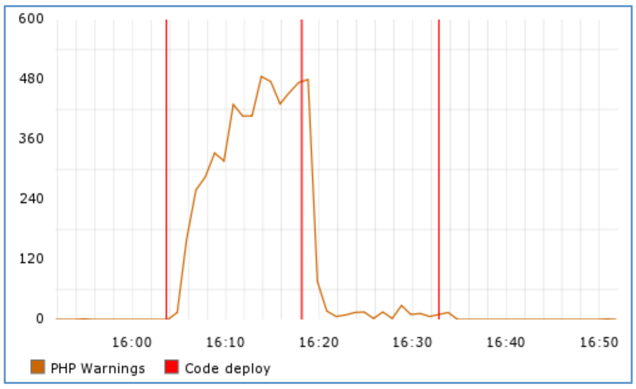


Figure 17 - Vertical lines marking code deploys on a time-series plot of errors

From participant interviews:

> *"At the very beginning of every #warroom, or really any outage or any anomaly, we're looking for correlation. A lot of our tooling ... In fact, many outages, or rather the most discrete possible event ...looking for correlation is something like a code push. In our tooling, we have a big vertical line that says, 'this is this event, at this time' and then any trends afterwards are really easy to correlate at that moment." (InfraEng5)*

> *"We've optimized for introspection for that kind of change in a big way, so that makes that convenient and easy. All of those factors make that by far the first thing we look for. It's the easiest thing to do." (InfraEng5)*

> *"If there is a vertical line and then a trend, you go back to the line, you can click the link, and you can then just look at what changes are visible, as in diff form. There's not of other changes so immediately visible and so explicit, the change in variables. The signal you get from that is very high." (InfraEng5)*

> *"At Etsy, it's very easy, because the first place I go, when people report things are breaking is, to the dashboard's index page. The pushes are already there, so you don't have to make that conscious decision. 'I want to look for a push that could have broken it.' It's like the graphs spiking or hockey sticks. 'Is there a line at the bottom of the hockey stick?' If yes, then take a look at what changed. Yeah, I think it's very close to the top of, 'I need some more information about what's going on, so who changed what?' (ProdEng2)*

Verbal reports are not the only supporting pieces of data on this heuristic. All of the dashboards that participants accessed had deploy lines enabled on the ones they loaded in their browser. Only InfraEng1 *disabled* the deploy lines on the API components performance dashboard at 01:25pm, reporting that once it was clear there were no deploys to be implicated in diagnosis, reporting that removing them from graphs made analysis easier.

Heuristic #2 – Time To Go Wide

Participants were asked about their perspective when discovering (as in this case) that there is *no* apparent correlation between a code change and the degradation they were seeing, which meant that their first diagnostic check was unfruitful. Their responses implied that their diagnostic directions were much more uncertain in those scenarios.

> *"I don't know. That means, I don't know, you're just starting from nothing. You have no… it's harder to have any kind of working theory." (ProdEng1)*

> *"That's when it gets scarier. It goes from feeling like 'oh something's broken, we can solve this'… to like: 'no idea what's wrong'. We could be, like, trying to diagnose this for days." (InfraEng3)*

> *"I think we had ruled out or at least got confidence that it wasn't a code change early on… and at that point then all bets are off." (InfraEng2)*

> *"If something didn't change, then I don't know where an issue would come from." (ProdEng2)*

> *"Another difficulty is when it's not code-push related. That throws open the door on possible ... what are the other possible causes? Or what else could be involved here?" (InfraEng4)*

ProdEng2 reflected about what the process they take in diagnosis:

> *"My brain makes a breadth first search, on ideas for what could be causing an issue. Then you try to evaluate them all, as quickly as possible. You think about, what could be failing? You think about, what is the quickest way for me to validate, whether or not that thing's failing? If that's not failing, cross it off the list. Then you find a little nugget, right?"*

> *"<InfraEng1> found the sidebar, and we all switched from breadth first to depth first, into the sidebar. We all jumped into the sidebar thread to figure out, what was going on there. Then we all hit dead ends, and switched back to breadth first, right?"*

Heuristic #3 – Convergent Searching

Once it becomes clear to engineers that heuristic #1 is proving to be unfruitful (i.e., a code change cannot be clearly implicated as contributing to the adverse behaviour) and that "all bets are off" and the *door is thrown open on other possible causes*, two main patterns emerged during this study:

a) Look to confirm/disqualify a specific and past diagnosis that comes to mind by matching signals or symptoms that appear similar.
b) Look to confirm/disqualify a general and *recent* diagnosis that comes to mind by matching signals or symptoms that appear similar.

#3a

A salient thread of diagnostic direction specific to InfraEng2's strategy during the event was explored during the interviews. While InfraEng2 did declare early in the timeline of the event to move the response dialogue from #sysops to the #warroom channel, the first diagnostic-related utterance InfraEng2 makes is a request for a specific piece of information:

```
[01:19:49 PM] < InfraEng2 > did some shops get frozen?
```

This query is notable in that it appears without referencing any relayed observations elsewhere in the transcript, and there were also no signals in the dashboard accesses or production site requests made by InfraEng2 to suggest frozen shops could be contributing to the issue.

Indeed, this direction (frozen shops) also appeared surprising to others in the channel – the first response to this query was:

```
[01:20:03 PM] < ProdEng3 > why would that affect this InfraEng2?
```

When asked about the origin of this diagnostic direction, InfraEng2 had an explanation.

> *"I've seen that probably four or five times over the time that I've been here. And you can see this big error thing and a lot of people get involved look into it and eventually find that they're all for either a specific listing or specific shop's listings."*

> *"And it was just kind of like, trying to think well: what kind of fits…looks similar to this that we're currently experiencing?"*

When asked later about the level of difficulty in diagnosing those previous events (where the primary trigger for the behaviour was when a shop's listings were frozen by the Trust and Safety group) InfraEng2 was able to recall vividly the struggle to diagnose them.

> **Investigator**: *"And when you've looked at those - experienced that sort of failure mode before, what was the level of difficulty in diagnosing it in the past?"*
>
> **InfraEng2:** *"Quite high, because it matched the same kind of symptoms we are seeing where there didn't seem to be an elevated number of errors in Supergrep there was no kind of first order available stores of data that said this, it was kind of like when we seen this particular thing before there it would be a particular graph on the deploy dashboard that would increase, which is errors that the users are experiencing, but that wouldn't be reflected anywhere else. It's one of the very first graphs on the deploy dashboard."*

Following InfraEng2's dialogue throughout the rest of the transcript as part of the cued recall during the interview, they recalled continuing to have this hypothesis in mind, and taking multiple steps to validate it.

When asked about what brought them to look in the logs specifically for errors relating to shop data (as shown in the narrative, at 1:21:01) they recalled:

> *"It felt like it was similar -- it felt like it was on the same path."*

When asked what brought them to share a screenshot of a graph in IRC (at 1:23:08), they recalled:

> *"I think this shows that it's definitely related. I think that at this point, I'm clear that it's very clearly related to the GetShopCards endpoint because were suddenly seeing huge increase in request time at this point I don't know if I have any firm grasp as to why that is suddenly elevated."*

InfraEng2 recalls feeling certain about their hypothesis that the issue is related to the retrieval of shop data, and when asked what brought them to re-state this at 1:24:07, they remark:

> *"I think at this point I'm just trying to get everyone in #warroom… to make everyone aware that this is the source, that in my mind there's no question about what this is, that this is where we should all be looking into now."*

As other engineers state other (potentially competing) hypotheses, ProdEng1 asks in channel:

```
[01:24:44 PM] < ProdEng1 > something wrong with the featured
shop?
```

And InfraEng2 replies:

```
[01:25:38 PM] < InfraEng2 > ProdEng2: yeah, I wonder if it's
frozen?
```

When asked about how this exchange (and others in the remaining parts of the transcript) could signal what their diagnostic direction was, InfraEng2 remarked:

> *"Right - I'm kinda fixated at this point, I think. (laughs)"*

"Because I've seen this before with the frozen shops kind of just doing weird things that no one can clearly point to straight away. I keep finding correlating evidence that says it's related to shops and I don't know if this point whether or not we've identified that it's a single shop or not, I can't recall, but that request was the only request that was returning 400s."

Both the remaining dialogue in the transcript and answers to questions in the interview supports InfraEng2 continuing to validate their initial hypothesis, and that a potential rationale for the fixation was the level of difficulty in resolving issues in the past that had similar symptoms.

The hypotheses that implicated both CDNs and Wordpress as causal also share this common theme: that once a code change can be *exonerated* as contributing to the observed behaviour and the diagnostic search space needs widening, a promising next step would be to match observations to those stored in memory.

ProdEng2 reflected a succinct rationale in the interview:

"Usually when I see symptoms that look familiar, I immediately check on the things that failed in the past."

Based on the process trace, there is a strong inference drawn that the criteria for events that come to mind are those that were remembered as either surprising or difficult to diagnose.

#3b

After the team had narrowed their diagnosis to the behaviour being related to the sidebar of the homepage, and then to shop data retrieval, and then to a specific blog post authored by an Etsy employee, the complete understanding of the underlying fault mechanism (*how* it was happening) still remained.

Before the blog post is taken down, a dialogue began to emerge in the channel focusing on understanding this pathology:

```
[01:35:39 PM] < InfraEng1 > varnish was queuing us?
```

At this point in the event, there are no requests found in the dashboard access logs that could be seen as informing this hypothesis, nor are there any previous mentions of Varnish or queuing in the transcript. When asked in the (pilot) interview about this, InfraEng1 recalled feeling certain about what the mechanism was, without any supporting evidence to validate the hypothesis.

When asked about the origin of this hypothesis, InfraEng1 recalled recognizing symptoms that also had appeared in recent events.

"Yeah. Like as soon as we saw timeouts that was the, sort of the first thing that's like, "Oh this is, is this like the queuing problem?"

These prior and recent events had a level of difficulty in diagnosis that was higher than the homepage degradation event.

"Well, so the initial, I mean, the first few times we saw the Varnish caching those were very difficult because we weren't used to that and we hadn't identified the patterns yet. And so the newness of it and unexpected behaviour where we were…like the server was entirely fine not serving any bad responses, but the Varnish cache was queuing those in an unexpected way and

> *that was surprising...and difficult to diagnose. We actually hit that same problem I think three times in the period of a week...because it kept happening in slightly different unexpected ways."*

At the beginning of the pilot interview with InfraEng1, when asked about whether they could recall the event, given the date and terse description, InfraEng1 guessed at which event it was by describing a fault mechanism specifically:

> **Investigator**: "*I am going to ask you some questions around an event that we had on December 4th. It was a homepage degradation. Do you remember that one?"*
>
> **InfraEng1:** *"I believe this would be related to varnish, caching multiple consolidating queries requests and queuing them up. Is this the one we're talking about?"*

InfraEng1's responses in the interview followed this theme. Many answers to questions were placed in the context of this fault mechanism.

With respect to the hypothesis generation in this event, these heuristics align well with (Woods & Hollnagel, 2006):

> "Performance at diagnostic reasoning depends on the ability to generate multiple plausible alternative hypotheses that might account for the findings to be explained. Observing anomaly response in action reveals that people do not call to mind all plausible alternatives at once. Rather, context, new evidence, and the response of the monitored process to interventions all serve as cues that suggest new possible explanations." (p. 77)

Heuristic #4: On Waiting For Tests To Pass

During the analysis phase of identifying coordinative episodes, an interesting juncture stood out: when it was time for the engineer deploying a code change during the event (in an effort to help resolution or minimize the impact of the degradation) to wait for the automated tests to finish, multiple times other engineers stated opinion that they shouldn't, and "push on through" the changes to production.

Given the role of automated testing in normal operations (to give the engineer deploying changes confidence that unimagined parts of the codebase would not be adversely affected by the given change) this made for an unintuitive paradox during the study. During an outage, deploying a change that complicated or added to the existing faults was a possibility. However, because there is perceived time pressure, maybe the engineers involved favoured not waiting for tests in order to resolve or contain the issue more quickly?

A simple survey was constructed and distributed amongst all engineers at Etsy. The questions were:

1. Before I deploy ANYTHING during an outage/degradation scenario, I will ask others if I should wait for tests to finish, or push on through. (yes/no)
2. When doing a CONFIG deploy during an outage/degradation scenario, on a scale of 1-5, where:
    a. 1 = I NEVER wait for tests to finish, I rely on feedback from others on my code change before deploying.

b. 5 = I ALWAYS wait for tests to finish, I don't care how much time pressure there is.

The results of question one were: 29 Yes, 3 No. (n=32)
The results of question two can be seen in Figure 18.

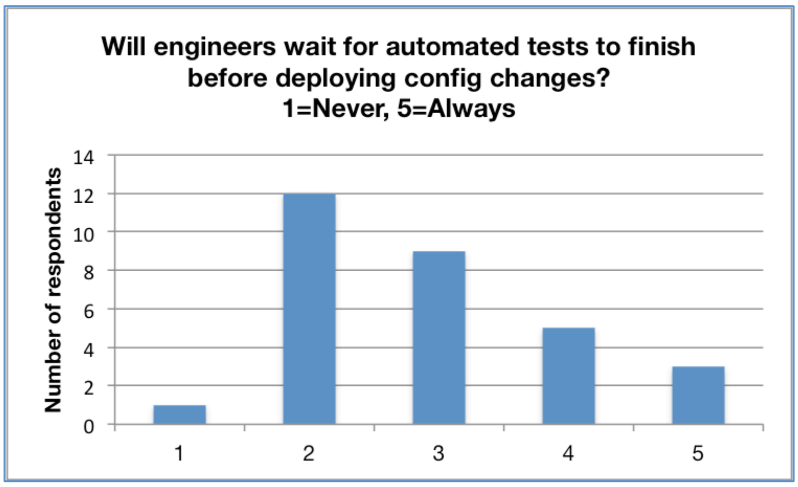


Figure 18 - Survey results: waiting for automated tests to finish

Some follow-up discussion with one of the respondents about the questions helped to provide more context on how they viewed the choice to wait or not:

> *"I think the two factors that affect my decision are the scope of the change and the scope of the outage. If I'm pushing a revert (vs a new patch), I have much more confidence that the tests will pass, since the reverted code has already been live (not 100%confidence, but high.) A new patch introduces more risk, especially unintended failures outside of where the patch is. I'm more likely to wait for the tests in that case. (EngS)*

This context and the survey results help validate that the rule-of-thumb of not waiting for automated tests to finish and instead relying on peer review for gaining confidence on code changes (at least CONFIG changes) is not limited to this particular event, but is more prevalent. This finding would align well with Hollnagel's examples of "Work-Related ETTO Rules" (Hollnagel, 2009) such as:

- `It has been checked earlier by someone else' - meaning that the correctness of the code change has been validated by a peer review
- `(Doing it) this way is much quicker' - meaning that a saving of time will happen by not waiting for the tests to finish

Abductive Reasoning

Given the heuristics that have been identified, an interpretation of the diagnostic activity is that it demonstrates that *abduction* is the primary mode of reasoning in events such as these. Using a frame proposed by Woods, we can place the observations engineers made (via the process trace) as D, and the hypotheses generated (CDN, network, Wordpress, database schema changes, frozen shops, etc.) during the event as H (Woods, 1995a):

- Step 1: D=collection of observations, givens, facts
- Step 2: H explains D, that is, H, if true, would account for D

- Step 3: No other available hypothesis explains D as well as H does
- Conclusion: Therefore, H is probably true.

In each of the hypotheses generated (Figure 16), interviews with participants corroborate this mode and process of reasoning.

DISCUSSION

The goal of this study was to explore the response to an Internet service outage event via the actions and communications that took place between software engineers as they attempted to restore service, in an effort to identify any "rules of thumb" that are used.

Heuristics were indeed identified during the course of analysing the data. The process trace constructed from externalizations in the IRC transcript and the system logs of engineer's behaviour (dashboard, site access, staff directory, etc.) provided salient cues that could be probed during interviews and result in strong inferences about what heuristics were "in play" during the event.

Along the way to identifying these heuristics, the necessary construction of process traces highlighted phases of reasoning and activity that aligns well with the anomaly response model found in Figure 1 and discussed in the Literature Review. The findings (and supporting observations) of this case study show more than just an alignment to the model shown in Figure 1, they augment the dynamic relationships between the three primary phases of *anomaly recognition, (re-)planning,* and *diagnosis.*

The activities did not follow any sort of simplistic directionality as one might interpret from the model in Figure 1, such as clockwise or counter-clockwise. Rather, observations validated the interrelationships between the phases in the model, and in some cases amplified the importance of specific relationships. Notable examples of this included:

- **Safing was a significant part of the response.** The primary goal of the coordinative episodes identified was to reduce the untoward effects of the degradation, to "buy" more time for diagnosis of the underlying mechanism, and potentially collect new or validating diagnostic data. In other words, engineers worked together to bring the site's functionality to a less-degraded state can be seen as a *therapeutic intervention.* (Woods & Hollnagel, 2006)
- **Diagnostic searches followed anomaly recognition in a cyclical pattern.** All hypotheses externalized during the diagnosis phase were followed by a validating (or not) anomaly detection search.
- **"Monitoring for success" followed each "corrective response" action.** Each of the coordinative episodes that resulted in an action was followed by a collective confirmation of success (or not) of the action. The disabling of the sidebar component entirely was followed by multiple confirmations that the errors stopped appearing in the logs. The re-enabling of the sidebar and disabling of the 'more from the blog' module was followed by reconfirmation of no errors and no degradation, via graphs and other relayed observations.

Interestingly, the coordinative episodes also follow the deeper enumeration of 'corrective responses' that Woods describes as the "four basic modes of response" within the '(re-)planning' phase (Woods, 1995a):

- **Mitigate consequences.** The initial coordinative episode began quite early in the response; ProdEng2 led the action taken to disable the sidebar when it was implicated in the pathology of the fault, even though the underlying mechanism of its relationship was unknown at that point.

- **Break propagation paths.** All of the coordinative episodes centred on removing the fault-producing code execution path from the primary functionality, the activity feed for signed-in users.
- **Terminate source.** After the diagnosis narrowed to a problematic blog post, it was unpublished, making it impossible for it to be referenced by executing code.
- **Clean-up after-effects.** The final state of the environment on the day of the outage was: the sidebar was fully enabled, the blog post was unpublished, and the 'more from the blog' module was left disabled. Over the following two days, ProdEng2 and ProdEng3 made multiple changes to both the API and the sidebar modules to gracefully handle the errors seen during the event.

As an aside (and perhaps just as important) is the validation of process-tracing as a promising and powerful approach in this field, due in part to the rich data sources found in digital artefacts and tools available to analyse this data efficiently.

Anomaly response and dynamic fault management are topics in this field that deserve greater attention, and there are a number of avenues yet to explore, and process-tracing should strongly be considered as an approach in future studies.

This study brought a great deal of focus and attention to the setting of Internet service software operations. If there is one regret, it is that the scope of this master's thesis cannot possibly follow all of the potential threads of investigation found along the way.

Perhaps unsurprisingly (to this investigator, anyway) more tempting questions emerged as the research unfolded, almost a "thematic vagabonding" phenomenon, in and of itself.

These still-unanswered questions could provide potential future directions, and they include:

- What heuristics might be in play as operators need assistance in expertise they do not have? In other words: what signal(s) would bring them to call for help or assistance from others?
- Many of the heuristics found were in diagnosis. Are there any rules-of-thumb used in the actions taken in response itself? (It could be surmised that the *graceful degradation* responses found in this case do qualify as heuristics, but there is not enough support in this study to make that claim.)
- How does length of experience in the domain influence the strategies operators take in anomaly response? (Klein, Hoffman 1992) suggests that strategies are not a definitive difference between novice and expert – can that finding be replicated in this environment?
- How does the organizational position of participants influence the anomaly response activities? In other words: do managers think differently about diagnostic directions, plans and re-planning of actions to take, etc. that do not correlate to the length of their experience?

It is hoped that these questions and others could be taken up in further studies, and that this research could assist in any way.

Additionally, the visualizations of the process tracing data were instrumental in enabling the investigator to discover new patterns (or threads of inquiry) than were initially expected, as well as falsifying assumptions and hypotheses held at the beginning of the study. Visualizations such as these have the potential to assist in not only exploration of events post-hoc, but also provide materials and new directions for training and education.

CONCLUSIONS

Engineers use heuristics in the process of resolving Internet service outages and degradations.

The *first* heuristic found in this study is: initially look for correlation between the behaviour and any recent changes made in the software.

The *second* heuristic found in this study is: upon finding no correlation with a software change, widen the search to any potential contributors imagined.

The *third* heuristic found in this study is: when choosing a diagnostic direction in a wide set of potential problem areas, reduce it by focusing on the one that most easily comes to mind, either because symptoms match those of a difficult-to-diagnose event in the past, or those of any recent events.

A *fourth* heuristic found in this study is coordinative in nature: when making changes to software in an effort to mitigate the untoward effects or to resolve the issue completely, rely on peer review of the changes more than automated testing (if at all.)

The primary limitation of this research is one that is found in all process-tracing approaches, which is that the inferences made about which cognitive strategies are in use can always benefit from more (and more diverse) data on externalizations. Expanding the volume of externalization data likely plateaus in its inferential value, however. The challenge is that such a plateau is not apparent.

While focus of this study was to identify any heuristics in use by engineers in anomaly response scenarios, there is a higher-level and perhaps *meta* conclusion, which is:

> The successful diagnosis and resolution of Internet service outages and degradations rely primarily on the expertise and tacit knowledge of engineers not only familiar with the software system(s) involved, but also with the diverse ways in which they can adversely behave.

To be sure, the heuristics identified (and observations made) in this study may prove to be very useful in the (re)design of tools intended to assist engineers in anomaly response. Indeed, this is the explicit intention.

However, in fields such as software operations, it can be too easy to assume that the success of business-critical systems are owed solely to the speed, algorithmic prowess, and *preventative* design of automation. This study joins many others in the fields of human factors, cognitive systems engineering, and systems safety that not only provide ample falsification of this perspective, but asserts a different view: that the greatest sources of success in automation-rich environments are (perhaps ironically) the adaptive capacities of *human* cognition and activity, not the pseudo-intelligence of software.

APPENDIX A – INITIAL CODING SCHEMA

The initial codes with descriptions and examples can be seen in Table 2. These codes were developed for the IRC transcripts and were focused on identifying salient points to discuss in the pilot interview.

Table 2 - Initial coding schema

Code	**Description**	**Example**
Asking for information, for diagnosis	A request for information, aimed at either specific individuals, or the entire channel, to aid diagnosis.	*"Has something happened to that endpoint?"*
Relaying observations	The volunteering of an observation of the system's behaviour or state	*"Also, the API error graph has spiked."*
Asking for clarity on a relayed observation	An attempt to repair the understanding of an observation made by someone else	*"What does that line mean in the graph?"*
Giving evidence of observation	Linking to graphs, errors, code, or other artefacts to support the observation relayed	*"Wait CPU per request spiked, look here: <URL to graph>"*
Suggesting a plan of action	A suggestion of an action to take, in the form of a question or statement, with the goal of either resolving an issue, safing, or interrogating the system for more information.	*"Should I shut that off?"*
Non-observation clarification	An attempt to clarify an earlier statement or repair understanding, not observation-related.	*"Eng1: just to be clear, you don't want me to push this config change to staging?"*
Stating a plan of action	A declaration of what action someone is intending to do, or is already doing.	*"I'm prepping a diff to turn off the sidebar and turn the blog back on."*
Asking about a plan of action	Asking others about what actions they're taking, or actions they are about to take.	*"Can we hold off on doing that?"*
Asking others for feedback on plan of action	An explicit request for feedback or "sanity check" on an action someone is about to take.	*"I can turn that off in isolation. How do folks feel about that?"*
Giving feedback on a plan of action	Feedback given on a plan of action, whether or not the feedback was requested.	*"That looks good, go for it."*
Confirming facts or state of process	Confirming or validating an understanding of the state of the event, observation that someone has made, or other factual data. It can also be the confirmation that an action has taken place.	*"Just the homepage is degraded."*
Asking about state of event	A request for the state of the event response, or asking for non-diagnostic information.	*"Have we communicated with the author of the blog post?"*

This coding schema evolved after the pilot interview into the schema below, with groupings of first-order codes and subcodes within them.

Diagnostic activity

- Asking for information for diagnosis
- Stating a diagnostic hypothesis
- Relaying observations
  - Giving evidence of observation
  - Asking for clarity on a relayed observation

Taking action

- Suggesting a plan of action
- Current or impending action
  - Asking other for sanity check
  - Asking about plan of action
  - Giving feedback on plan

Confirming facts or state or process

- Asking about state of the outage response

Finally, for use in the graphical representations in the study, the first-order code were collapsed into two main codes:

- **Diagnostic activity**
- **Taking action/response**

Reliability In Coded Data

In order to measure reliability of the coding system, another coder's results with the same document were used to find a percentage agreement in coding of 79.6%. This second coder was someone familiar with terminology but not familiar with the specific event.

Table 3 - Intercoder Percentage Agreement

Code	Correlates	Doesn't correlate	Total	Percent Agreement
Suggesting a plan of action	50	13	63	79.3
Stating a diagnostic hypothesis	58	34	92	63.04
Relaying observations	158	42	200	79.0
Giving feedback on plan of action	118	18	136	86.7
Giving evidence of observation	78	17	95	82.1
Confirming facts or state of process	159	41	200	79.5
Asking other for sanity check	20	7	27	74.07
Asking for information for diagnosis	100	21	121	82.6
Asking for clarity on a relayed observation	14	9	23	60.8
Asking about state of event	27	12	39	69.2
Asking about plan of action	75	6	81	92.5
	857	208	1077	79.6

Appendix B – Coordinative Episodes Coding Schema

Below is the entirety of coordination episode #1, as an example of the strategy taken with all identified coordinative episodes.

- **Suggestion of action/request for feedback** (SoA/RFF) – typically the initial state of a coordinative episode.
- **Feedback on action/artifact** (FoA/Art) – the acknowledgement that an action is suggested or artefact has been presented for feedback, and the feedback itself
- **Asserting state** (AS) – a form of 'closed-loop' communication wherein an actor reports the current state of the environment or impending actions they're taking

Time	Participant	Utterance	Code
1:19:10	ProdEng2	`should I shut that off?`	SoA/RFF
1:19:14	ProdEng3	`yeah try it`	FoA/Art
1:19:19	ProdEng2	`one sec, let me cook up a diff`	AS
1:21:33	ProdEng2	`folks,  gist to turn off the sidebar: <URL>`	SoA/RFF
1:21:42	ProdEng3	`ProdEng2: lgtm`	FoA/Art
1:21:43	ProdEng1	`lgtm`	FoA/Art
1:21:45	ProdEng2	`(waiting on try-config)`	AS
1:21:50	ProdEng3	`ProdEng2: dont`	FoA/Art
1:21:58	ProdEng2	`keke`	FoA/Art
1:22:34	ProdEng2	`in the config queue and pushing change`	FoA/Art
1:22:57	ProdEng2	`pushing to princess`	AS
1:22:58	EngA	`ProdEng2: is that turning on the sidebar?`	FoA/Art
1:23:10	ProdEng2	`EngA: turning "on" the sidebar kill switch`	FoA/Art
1:23:42	ProdEng2	`on princess`	AS
1:23:46	ProdEng2	`Should I push through tests?`	SoA/RFF
1:24:04	ProdEng2	`please confirm folks`	SoA/RFF
1:24:06	ProdEng2	`<URL>`	SoA/RFF
1:24:23	EngB	`ProdEng2: lgtm signed in`	FoA/Art
1:24:30	ProdEng1	`lgtm for me`	FoA/Art
1:24:33	EngC	`me too ProdEng2`	FoA/Art
1:24:34	InfraEng3	`lgtm signed in en-us`	FoA/Art
1:24:36	EngD	`ProdEng2: i say roll on through`	FoA/Art
1:24:42	ProdEng2	`going to prod`	AS
1:24:43	EngE	`samesies`	FoA/Art
1:25:37	ProdEng2	`homepage is back`	AS
1:25:40	ProdEng2	`config is on prod`	AS
1:25:41	ProdEng3	`confirmed homepage looks good on princess`	FoA/Art

APPENDIX C – PARTICIPANT TIMELINES

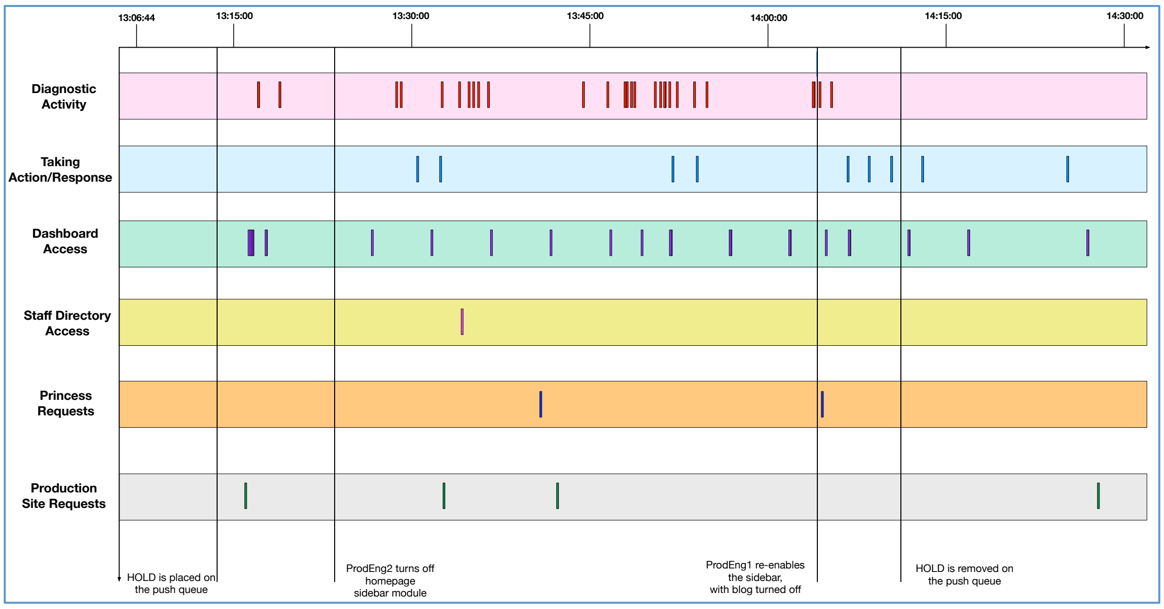


Figure 19 - Infrastructure Engineer 1 timeline

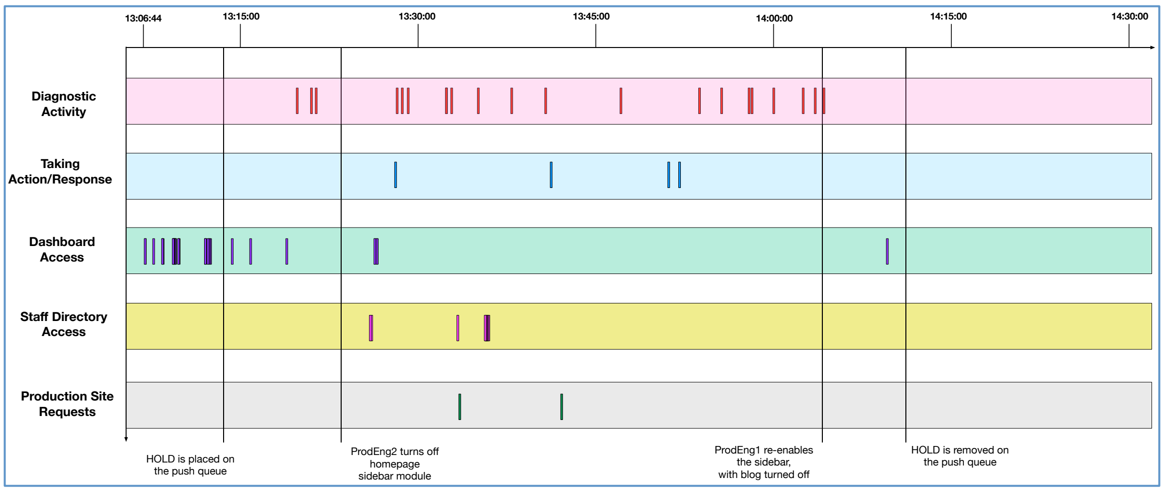


Figure 20 - Infrastructure Engineer 2 timeline

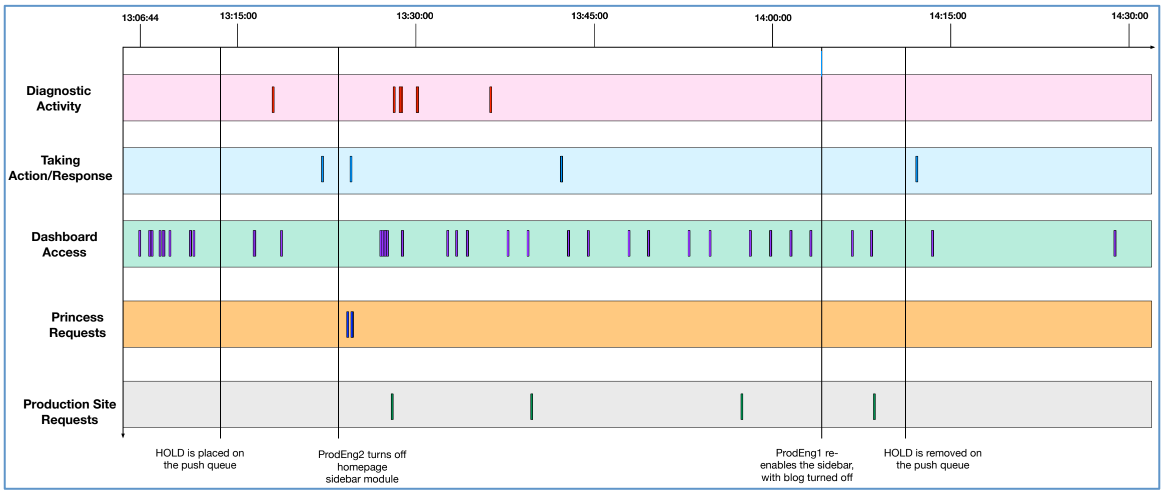


Figure 21 - Infrastructure Engineer 3 timeline

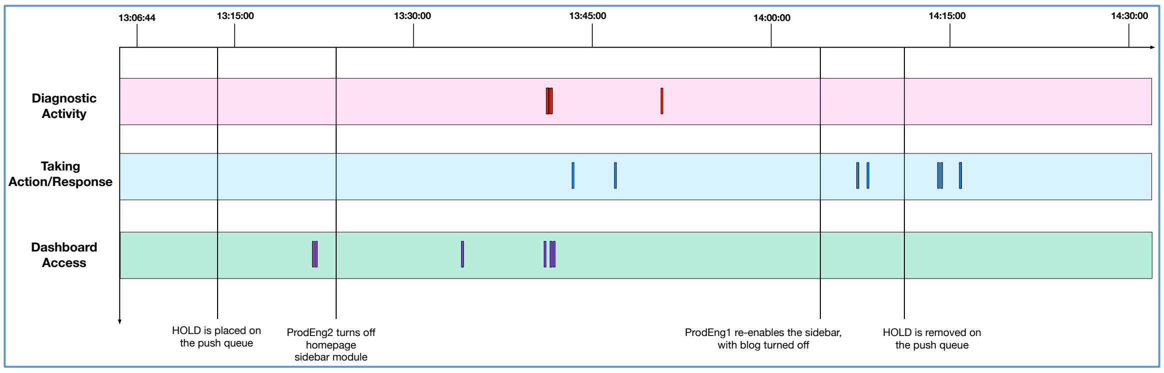


Figure 22 - Infrastructure Engineer 4 timeline

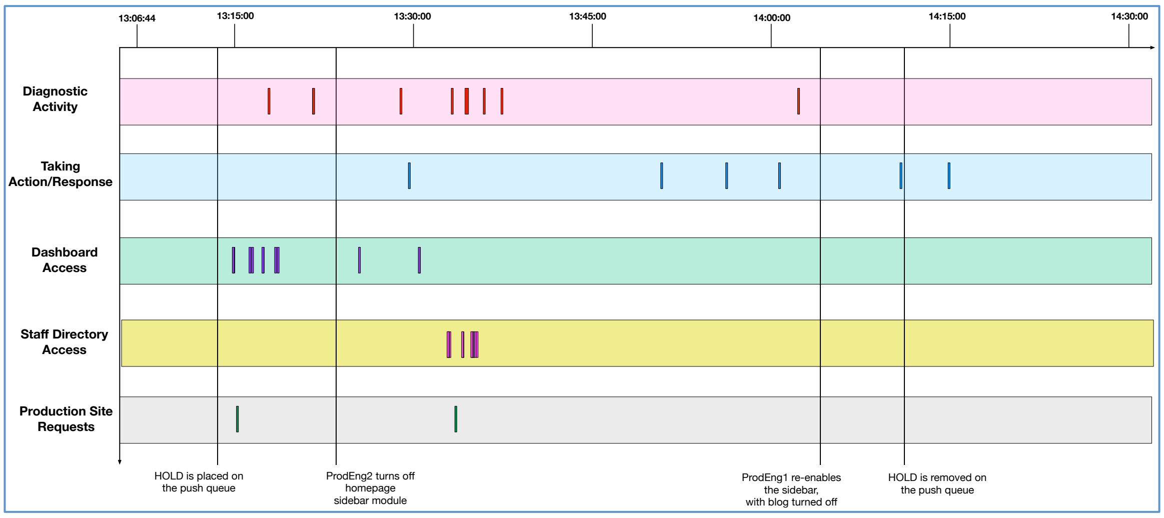


Figure 23 - Infrastructure Engineer 5 timeline

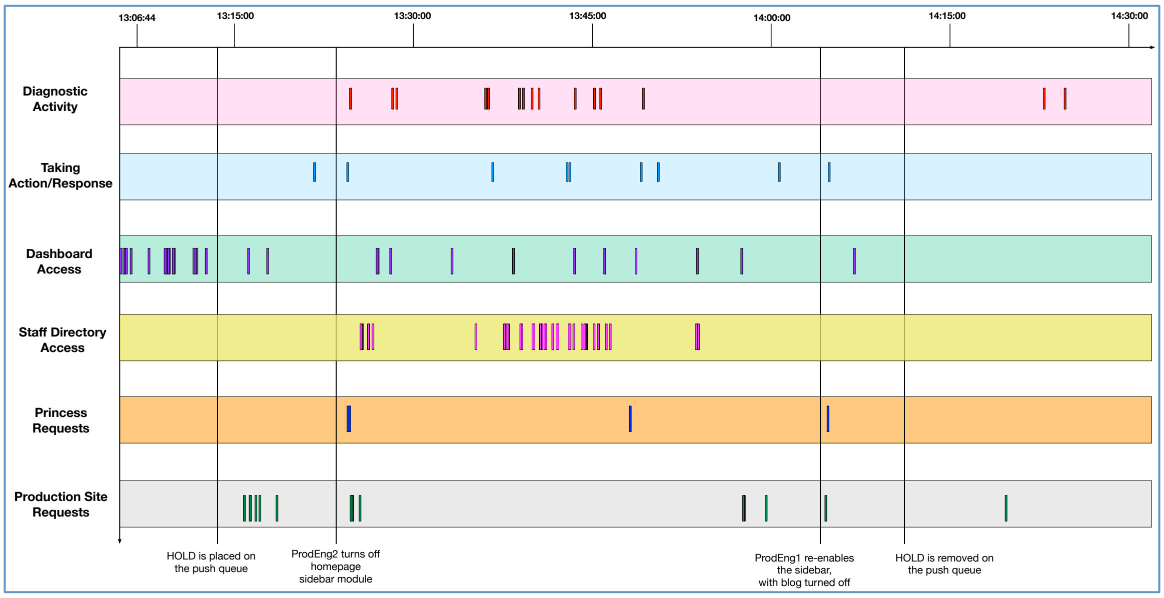


Figure 24 - Product Engineer 1 timeline

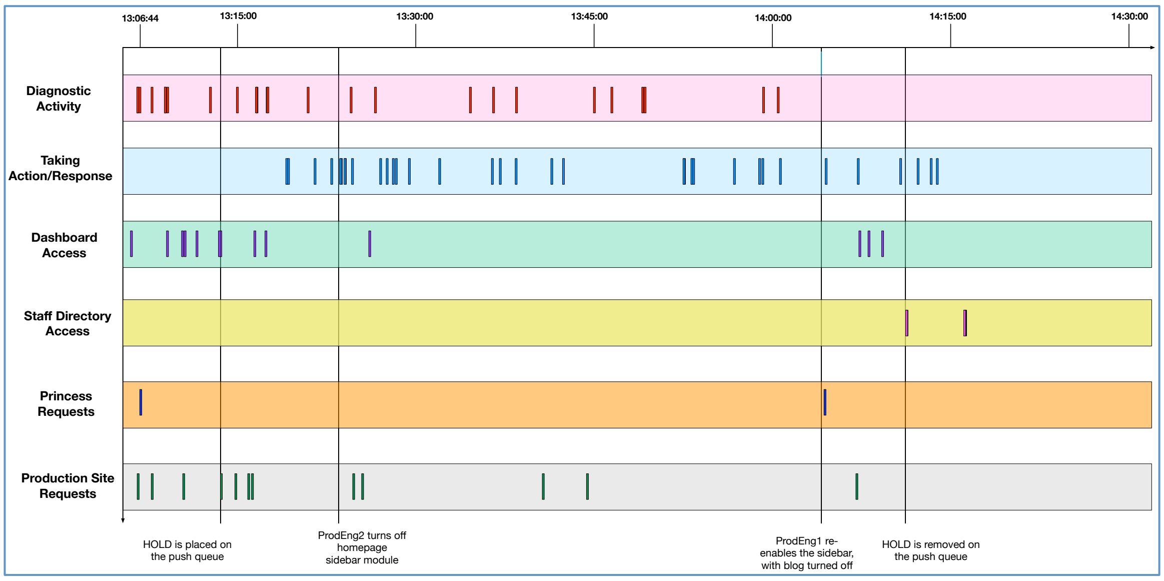


Figure 25 - Product Engineer 2 timeline

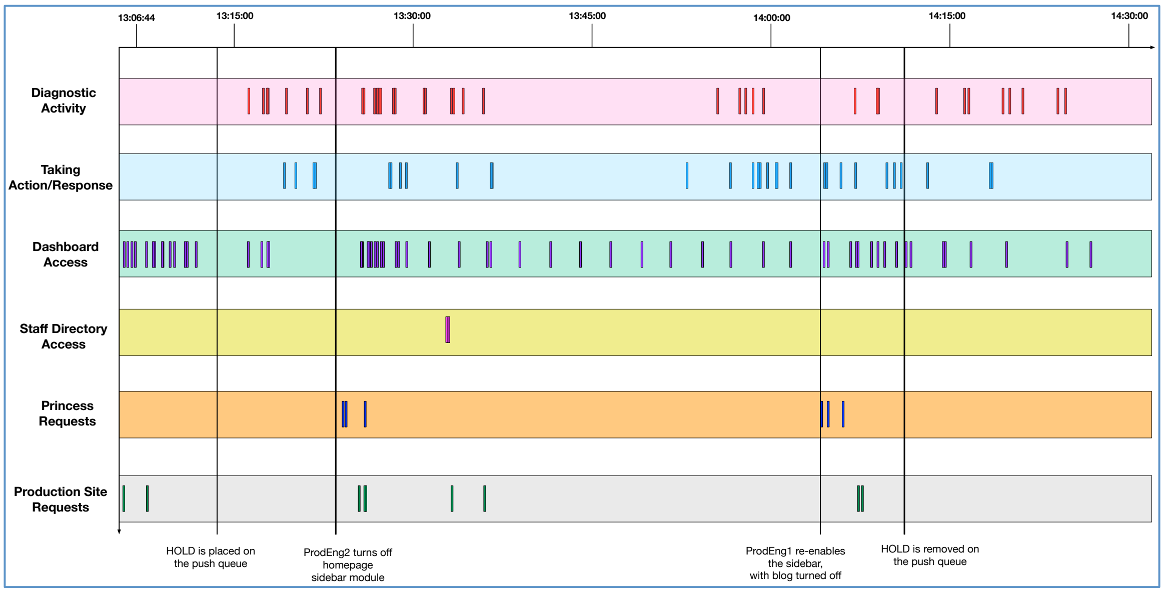
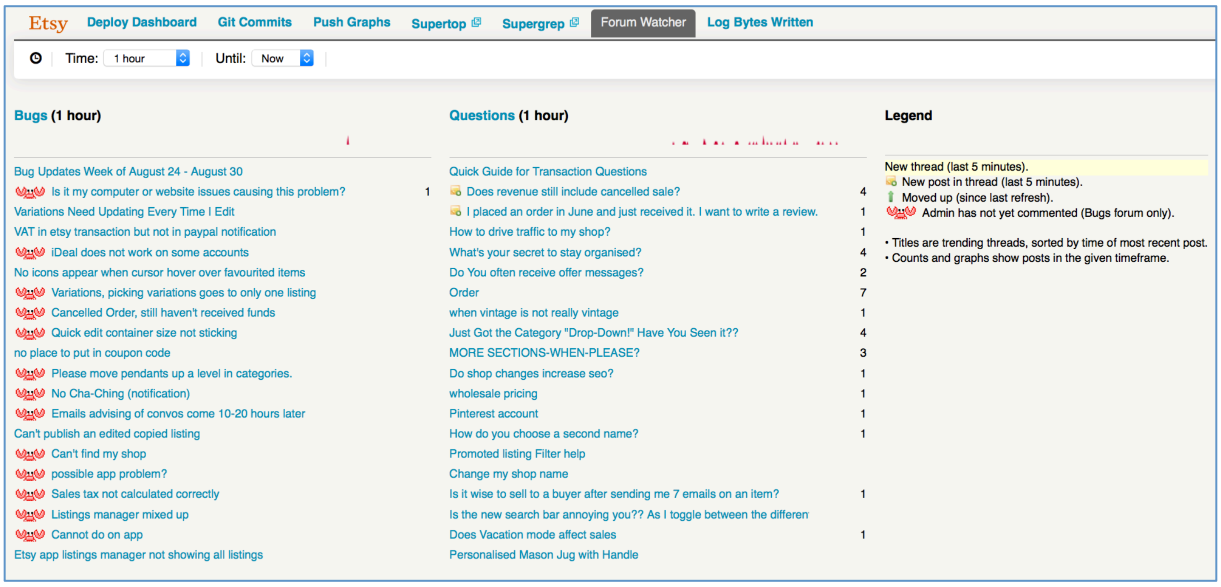


Figure 26 - Product Engineer 3 timeline

APPENDIX D – SCREENSHOTS OF DASHBOARDS

The following are screenshots of the dashboards engineers viewed during the outage. These are for illustration purposes only; the actual data and graphs shown represent *current* data, not from the day of the event.



Time: 1 hour | Until: Now

Bugs (1 hour)

Bug Updates Week of August 24 - August 30	
Is it my computer or website issues causing this problem?	1
Variations Need Updating Every Time I Edit	
VAT in etsy transaction but not in paypal notification	
iDeal does not work on some accounts	
No icons appear when cursor hover over favourited items	
Variations, picking variations goes to only one listing	
Cancelled Order, still haven't received funds	
Quick edit container size not sticking	
no place to put in coupon code	
Please move pendants up a level in categories.	
No Cha-Ching (notification)	
Emails advising of convos come 10-20 hours later	
Can't publish an edited copied listing	
Can't find my shop	
possible app problem?	
Sales tax not calculated correctly	
Listings manager mixed up	
Cannot do on app	
Etsy app listings manager not showing all listings	

Questions (1 hour)

Quick Guide for Transaction Questions	
Does revenue still include cancelled sale?	4
I placed an order in June and just received it. I want to write a review.	1
How to drive traffic to my shop?	1
What's your secret to stay organised?	4
Do You often receive offer messages?	2
Order	7
when vintage is not really vintage	1
Just Got the Category "Drop-Down!" Have You Seen it??	4
MORE SECTIONS-WHEN-PLEASE?	3
Do shop changes increase seo?	1
wholesale pricing	1
Pinterest account	1
How do you choose a second name?	1
Promoted listing Filter help	
Change my shop name	
Is it wise to sell to a buyer after sending me 7 emails on an item?	1
Is the new search bar annoying you?? As I toggle between the differen	
Does Vacation mode affect sales	1
Personalised Mason Jug with Handle	

Legend

New thread (last 5 minutes).
New post in thread (last 5 minutes).
Moved up (since last refresh).
Admin has not yet commented (Bugs forum only).

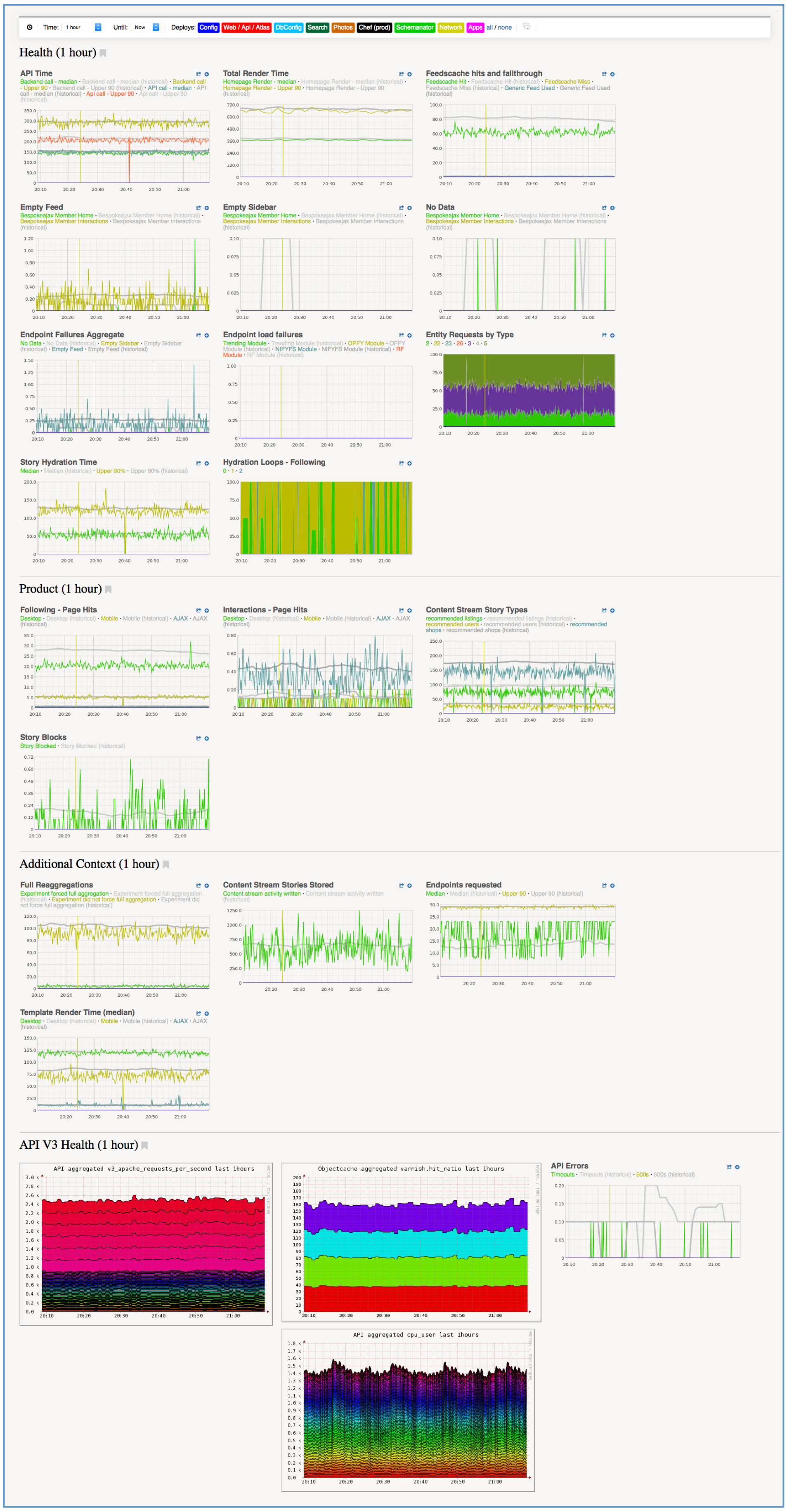
- Titles are trending threads, sorted by time of most recent post.
- Counts and graphs show posts in the given timeframe.

Figure 27 - Forums dashboard

The "Forum Watcher" dashboard (shown in Figure 27) highlights most recent discussion threads on Etsy.com's bug forums. Engineers typically view this dashboard as an outage or degradation unfolds in order to collect any reports from users on the site that could inform diagnosis or response.

Signed-in homepage performance dashboard

A screenshot of the signed-in homepage dashboard can be seen in Figure 28. This dashboard consists of graphs curated by the team that designed and built the signed-in homepage, which included ProdEng1, 2 and 3. What appears here are various indications of the feature's health, as well as metrics concerning the general health of the API server cluster and the caching hit ratio of the caching cluster.



Time: 1 hour
Until: Now
Deploys: Config Web / Api / Atlas DbConfig Search Photos Chef (prod) Schemanator Network Apps all / none
Health (1 hour)
API Time
Total Render Time
Feedscache hits and fallthrough
Empty Feed
Empty Sidebar
No Data
Endpoint Failures Aggregate
Endpoint load failures
Entity Requests by Type
Story Hydration Time
Hydration Loops - Following
Product (1 hour)
Following - Page Hits
Interactions - Page Hits
Content Stream Story Types
Story Blocks
Additional Context (1 hour)
Full Reaggregations
Content Stream Stories Stored
Endpoints requested
Template Render Time (median)
API V3 Health (1 hour)
API aggregated v3_apache_requests_per_second last 1hours
Objectcache aggregated varnish.hit_ratio last 1hours
API Errors
API aggregated cpu_user last 1hours

Figure 28 - Signed-in homepage dashboard

API Performance Dashboards

A common set of dashboards that were used in diagnosis extensively by InfraEng1, InfraEng3, and ProdEng3.

The dashboard has four modes, which can be selected from a drop-down menu found in the header of the page, as seen in Figure 29.

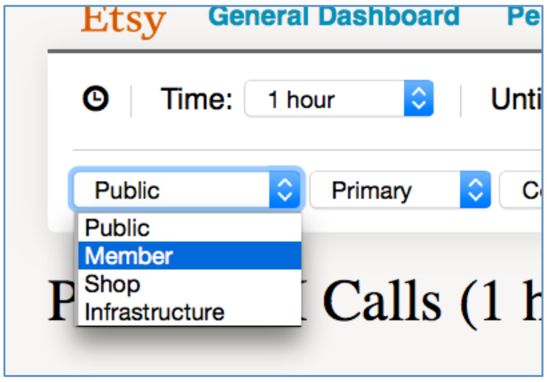


Figure 29 - API Component Performance Dashboard Mode Menu

In addition, a time period for the given graphs on the page can be chosen from the header, as shown in Figure 30.

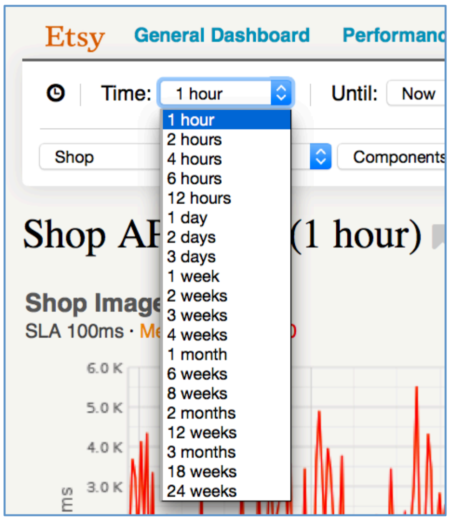


Figure 30 - Time period selection for dashboards

The *Public* mode displays time-series graphs for the median and upper 90th percentile of response latency of up to thirty (30) "public" API endpoints. These endpoints can be seen as generalized methods to obtain fundamental data about entities in Etsy's database. Examples are: shop information, seller profiles, data about listings for sale, etc. As it pertains to this study, the performance of the *Public_Shops_GetShopCards* method (which is referenced in both the IRC transcripts as well as interviews) would be found on this dashboard.

The *Member* mode of this dashboard was also seen during this event. Like the Public mode, this displays median and 90th percentile metrics for API calls that involve setting or displaying data specific to a user's experience with the application. A specific example would be the performance of the *Member_Homepage_Sidebar* method, which was also referenced in IRC transcripts and interviews.

In addition, some participants also viewed the *Shop* mode of this dashboard during the outage, although observations relating to it were not specifically externalized. A composite screenshot of these three modes, along with a guideline as to where a browser typically would limit the initial view of the page can be seen in Figure 31. These screenshots were taken after the event and are not intended to reflect the state of performance during the event. The *Public_Shops_GetShopCards* method metrics are highlighted in the blue box.

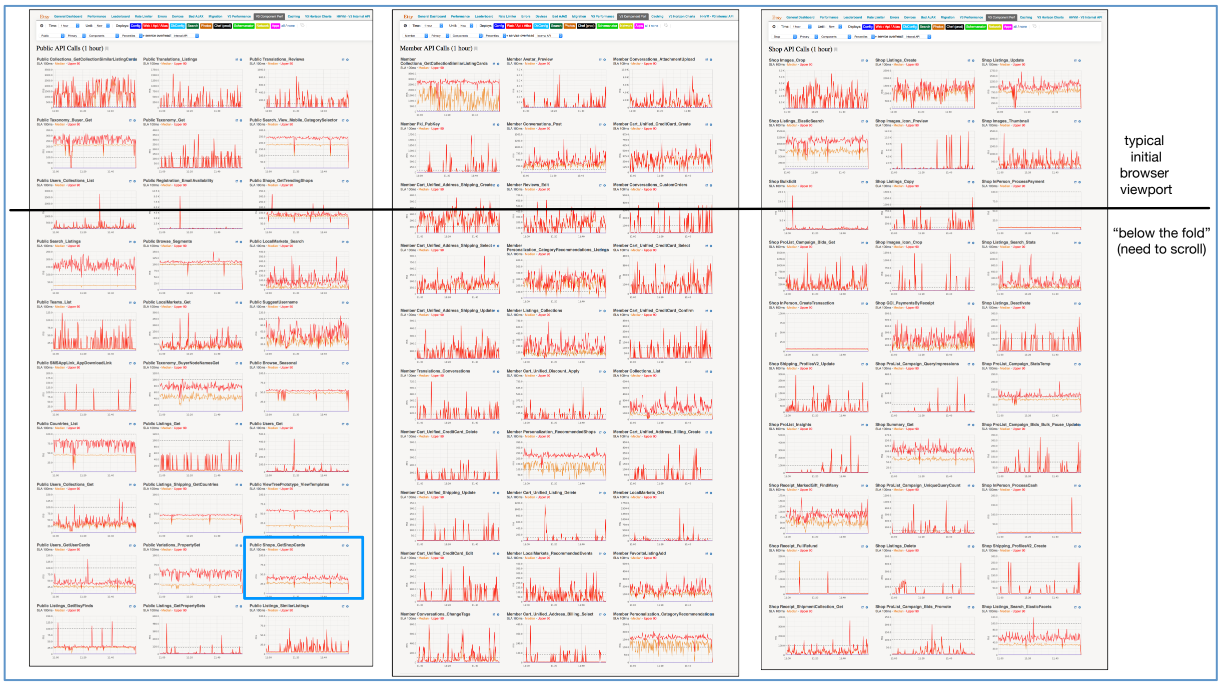
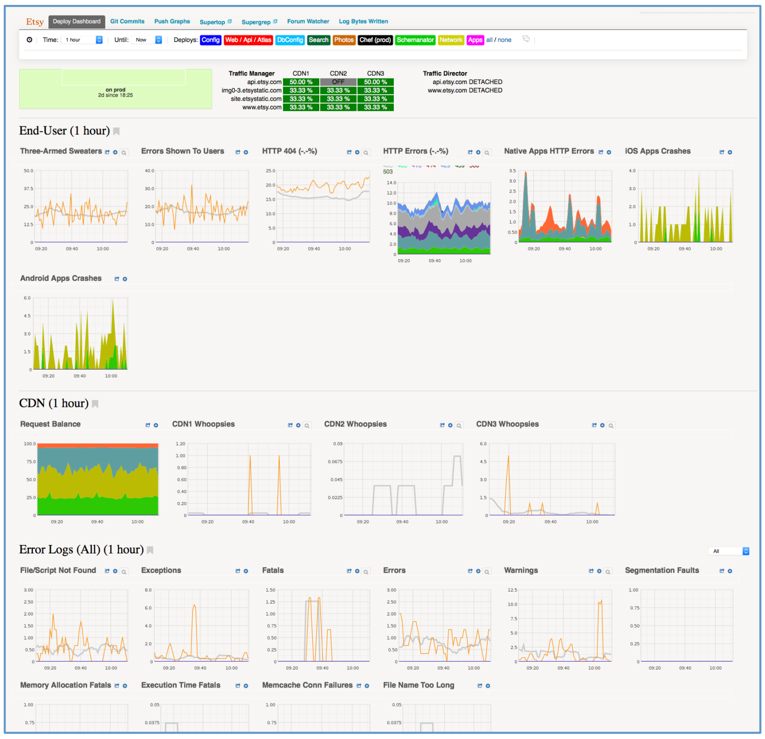

Public API Calls (1 hour)
Member API Calls (1 hour)
Shop API Calls (1 hour)
typical initial browser viewport
"below the fold" (need to scroll)

Figure 31 - The Public, Member, and Shop modes of the API component performance dashboard

Deploy dashboard

The deploy dashboard is a composite of time-series metrics and high-level status indicators that engineers will look to (amongst other places) when they are deploying new code to production. It is intended to give a wide-but-shallow look on business and systems metrics as a first view into anomalies that might arise.

The top portion of the deploy dashboard is seen in Figure 32, below. Not all of the dashboard can be represented here, as there are business-sensitive metrics as part of the dashboard.



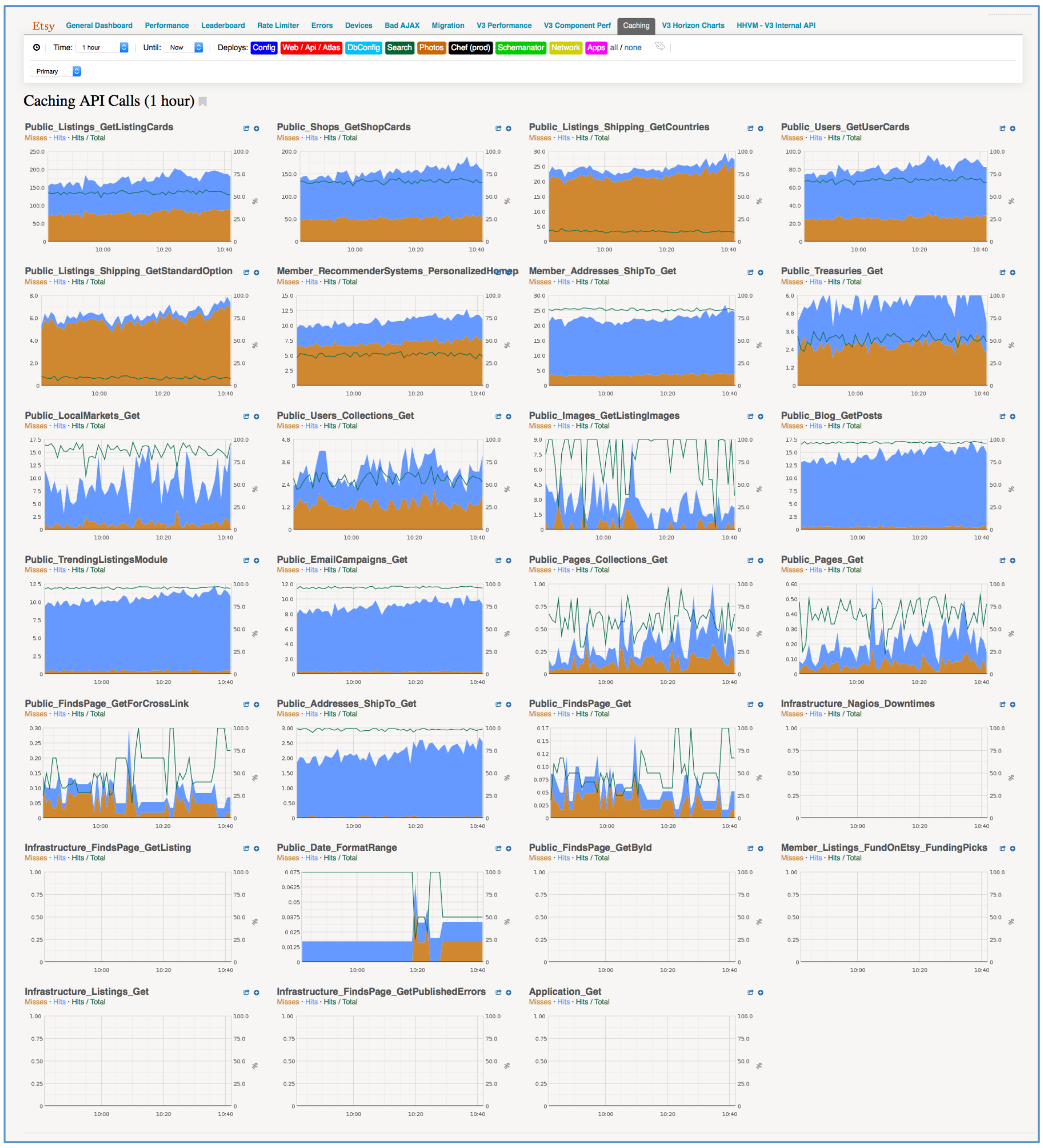
Traffic Manager	CDN1	CDN2	CDN3
api.etsy.com	50.00 %	OFF	50.00 %
img0-3.etsystatic.com	33.33 %	33.33 %	33.33 %
site.etsystatic.com	33.33 %	33.33 %	33.33 %
www.etsy.com	33.33 %	33.33 %	33.33 %

Traffic Director
api.etsy.com DETACHED
www.etsy.com DETACHED

Figure 32 - Portion of the deploy dashboard

API caching dashboard

The API caching dashboard displays the cache hit ratios of the caching cluster, seen in Figure 33.


Etsy
General Dashboard
Performance
Leaderboard
Rate Limiter
Errors
Devices
Bad AJAX
Migration
V3 Performance
V3 Component Perf
Caching
V3 Horizon Charts
HHVM - V3 Internal API
Deploys: Config Web / Api / Atlas DbConfig Search Photos Chef (prod) Schemanator Network Apps all / none
Caching API Calls (1 hour)
Public_Listings_GetListingCards
Public_Shops_GetShopCards
Public_Listings_Shipping_GetCountries
Public_Users_GetUserCards
Public_Listings_Shipping_GetStandardOption
Member_RecommenderSystems_PersonalizedHomep
Member_Addresses_ShipTo_Get
Public_Treasuries_Get
Public_LocalMarkets_Get
Public_Users_Collections_Get
Public_Images_GetListingImages
Public_Blog_GetPosts
Public_TrendingListingsModule
Public_EmailCampaigns_Get
Public_Pages_Collections_Get
Public_Pages_Get
Public_FindsPage_GetForCrossLink
Public_Addresses_ShipTo_Get
Public_FindsPage_Get
Infrastructure_Nagios_Downtimes
Infrastructure_FindsPage_GetListing
Public_Date_FormatRange
Public_FindsPage_GetById
Member_Listings_FundOnEtsy_FundingPicks
Infrastructure_Listings_Get
Infrastructure_FindsPage_GetPublishedErrors
Application_Get
Misses · Hits · Hits / Total

Time: 1 hour Until: Now

Primary

Figure 33 - API caching dashboard

Four other miscellaneous dashboards were viewed by engineers and used as prompts during interviews, but are not included here due to their large size and their minor role in the focus of the study.

Appendix E – Tools used in analysis

A number of tools were used in this research to transform, reduce, diagram, and otherwise analyse the data:

Dateutils collection

The *dateutils* collection of command-line tools (http://www.fresse.org/dateutils/) proved to be invaluable in this research. Because the process tracing approach involved collecting time-series data (such as logs) from a number of disparate and diverse sources, the date and time formats are rarely consistent. Add to this challenge the complication of different time zones (some of the data was recorded in UTC, whereas others were in EST) and much transformation and normalizing of the data on a single format (the Unix epoch format) was necessary. The *dateutil* tools (dateconv, dateseq, and dategrep, especially) made this task straightforward.

MaxQDA

MaxQDA (http://www.maxqda.com/) is a qualitative analysis software tool which proved to be very useful in coding the IRC transcripts and interviews, and made refining the coding schemas and annotating the documents very easy. None of the analysis was performed via any automated analysis features of MaxQDA.

Omnigraffle

OmniGraffle (https://www.omnigroup.com/omnigraffle) is a diagramming and digital illustration tool. Systems log data and timestamps from the IRC transcripts were imported into OmniGraffle, where they could then be highlighted and annotated and places on the timeline of the event.

Hindenburg Field Recorder App

The Hindenburg Field Recorder App (http://hindenburg.com/) is a high-quality and easy-to-use audio recording app for iOS. It was used to record interviews with participants.

Rev.com transcription service

Rev (http://rev.com) is an online audio transcription service used to transcribe the interview audio files into text. The transcriptions were reviewed alongside the audio in order to catch any mistakes in the transcription.

References

"About Etsy." *About Etsy*. Etsy, Inc., n.d. Retrieved Web. 30 Sept. 2015, from https://www.etsy.com/about.

Ahmed, W., & Wu, Y. W. (2013). A survey on reliability in distributed systems. *Journal of Computer and System Sciences*, *79*(8). http://doi.org/10.1016/j.jcss.2013.02.006

Bainbridge, L. (1979). Verbal Reports as Evidence of the Process Operators Knowledge. *International Journal of Man-Machine Studies*, *11*(4), 411–436.

Bergström, J., Dahlström, N., & Henriqson, E. (2010). Team Coordination in Escalating Situations: An Empirical Study Using Mid□ Fidelity Simulation. Journal of Contingencies and Crisis Management, 18: 220–230. doi: 10.1111/j.1468-5973.2010.00618.x

Bruns, A., Highfield, T., & Burgess, J. (2013). The Arab Spring and Social Media Audiences: English and Arabic Twitter Users and Their Networks. *American Behavioral Scientist*, *57*(7), 871–898. http://doi.org/10.1177/0002764213479374

Yatin Chawathe and Eric A. Brewer. 2009. System support for scalable and fault tolerant internet services. In *Proceedings of the IFIP International Conference on Distributed Systems Platforms and Open Distributed Processing* (Middleware '98), Jochen Seitz (Ed.). Springer-Verlag, London, UK, UK, 71-88.

Cook, R. I. (1998). How complex systems fail. Cognitive Technologies Laboratory, University of Chicago. Chicago IL.

Cook, R. I. (2010). How Complex Systems Fail. In J. Allspaw & J. Robbins (Eds.), *Web Operations: Keeping the Data On Time, 1st edition.* O'Reilly Media, Inc.

Crandall, B., Klein, G. A., & Hoffman, R. R. (2006). Working minds: A practitioner's guide to cognitive task analysis. Cambridge, Mass. : MIT Press.

Creswell, John W., and John W. Creswell. *Qualitative Inquiry & Research Design: Choosing among Five Approaches*. Thousand Oaks: Sage Publications, 2007. Print.

Cyber Monday. (n.d.). Cyber Monday. Wikipedia. Retrieved from http://en.wikipedia.org/wiki/Cyber%20Monday

Distributed computing. (2015, July 9). Retrieved July 12, 2015, from https://en.wikipedia.org/wiki/Distributed_computing

Dörner, D. (1980). On the Difficulties People Have in Dealing With Complexity. *Simulation & Gaming*, *11*(1), 87–106. http://doi.org/10.1177/104687818001100108

Ericsson, K. A. K. A. 1., & Simon, H. A. H. A. 1.-2. (1984). Protocol analysis : verbal reports as data. Cambridge, Mass. : MIT Press.

Flanagan, J. C. (1954). The critical incident technique. *Psychological bulletin*, *51*(4), 327.

Ford, P. (2013, October 16). The Obamacare Website Didn't Have to Fail. How to Do Better Next Time. Retrieved July 12, 2015, from http://www.bloomberg.com/bw/articles/2013-10-16/open-source-everything-the-moral-of-the-healthcare-dot-gov-debacle

Gilbert, S., & Lynch, N. (2002). Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *SIGACT News*, *33*(2), 51. http://doi.org/10.1145/564585.564601

Grisogono, A.-M. (2006). *The implications of complex adaptive systems theory for C2*. DTIC Document.

Haber, E. M., Kandogan, E., & Maglio, P. (2010). Collaboration in System Administration. *Queue*, *8*(12), 10. http://doi.org/10.1145/1898147.1898149

Hellerstein, J. M., & Stonebraker, M. (2005). Readings in Database Systems. Cambridge: MIT Press.

Hollnagel, E. (2009). The ETTO Principle: Efficiency-Thoroughness Trade-Off. Ashgate.

Huber, O., & Seiser, G. (2001). Accounting and convincing: The effect of two types of justification on the decision process. *Journal of Behavioral Decision Making*, *14*(1), 69.

Hutchins, E. (1995). *Cognition in the Wild*. MIT press.

Kaheman, D., & Klein, G. (2009). Conditions for Intuitive Expertise. *American Psychologist* (Vol. 64, pp. 515–526). American Psychological Association. http://doi.org/10.1037/a0016755

Kahneman, D., & Tversky, A. (1974). Judgment under Uncertainty: Heuristics and Biases., *185*(4157), 1124–1131. http://doi.org/10.1126/science.185.4157.1124

Kandogan, E., Maglio, P., Haber, E., & Bailey, J. (2012). Taming Information Technology: Lessons from Studies of System Administrators. Oxford University Press, USA.

Klein, G. (2006). The strengths and limitations of teams for detecting problems. *Cognition, Technology & Work*, *8*(4). http://doi.org/10.1007/s10111-005-0024-6

Klein, G. A. (1993). Decision making in action : models and methods. Norwood, N.J. : Ablex Pub.

Klein, G. A., & Hoffman, R. (1993). Seeing the invisible: Perceptual/cognitive aspects of expertise. In M. Rabinowitz (Ed.), *Cognitive science foundations of instruction* (pp. 203-226). Mahwah, NJ: Lawrence Erlbaum Associates.

Klein, G., Feltovich, P. J., Bradshaw, J. M., & Woods, D. D. (2005). Common ground and coordination in joint activity. *Organizational Simulation*, 139–184.

Knott, B. A., Nelson, W. T., & Brown, R. D. (2007). Assessment of a novel chat messaging interface for rapid communication in tactical command and control. Presented at the Proceedings of the …. http://doi.org/10.1177/154193120705101823

Maglio, P. P., Kandogan, E., & Haber, E. (2008). Distributed Cognition and Joint Activity in Computer System Administration. In *Resources, Co-Evolution and Artifacts* (pp. 145–166). London: Springer London. http://doi.org/10.1007/978-1-84628-901-9_6

Margetts, H. (2015). Citizens, Governments and Information in a Changing Digital World (pp. 1–35). Presented at the C2 Symposium. Retrieved from https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/417713/Prof_Helen_Margetts__University_of_Oxford.pdf

Miller, J. H., & Page, S. E. (2009). Complex adaptive systems: an introduction to computational models of social life: an introduction to computational models of social life. Princeton university press. http://doi.org/10.2307/j.ctt7s3kx

Neisser, U. (1976). Cognition and reality: Principles and implications of cognitive psychology. New York, NY, US: W H Freeman/Times Books/ Henry Holt & Co.

Nisbett, R. E., & Wilson, T. D. (1977). Telling More Than We Can Know - Verbal Reports on Mental Processes. *Psychological Review*, *84*(3), 231–259.

Patrick, J., & James, N. (2004). Process tracing of complex cognitive work tasks. *Journal of Occupational and Organizational Psychology*, *77*(2), 259–280.

Patterson, E. S. (1999, May). Computer-supported inferential analysis under data overload. In *CHI'99 Extended Abstracts on Human Factors in Computing Systems* (pp. 67-68). ACM.

Patterson, E. S., Watts-Perotti, J., Smith, P. J., Woods, D. D., Shattuck, L., Klein, G., & Jones, P. M. (1998, October). Patterns in cooperative cognition. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (Vol. 42, No. 3, pp. 263-266). SAGE Publications.

Performance Activity on GOV.UK: Web traffic. (2015, July 12). Retrieved July 12, 2015, from https://www.gov.uk/performance/site-activity

Phister, P. W., Jr. (2010). *Cyberspace: The ultimate complex adaptive system*. DTIC Document.

Rogers, Y. (1992). Ghosts in the network: distributed troubleshooting in a shared working environment (pp. 346–355). Presented at the CSCW '92: Proceedings of the 1992 ACM conference on Computer-supported cooperative work, New York, New York, USA: ACM Request Permissions. http://doi.org/10.1145/143457.143546

Schulte-Mecklenbeck, M., Kühberger, A., & Ranyard, R. (2011). A handbook of process tracing methods for decision research: A critical review and user's guide. New York : Psychology Press.

Simon, H. A. (1992). What Is an "Explanation" of Behavior? *Psychological Science*, *3*(3), 150–161. http://doi.org/10.1111/j.1467-9280.1992.tb00017.x

Thornett, J. (2015, June 17). The new normal for GOV.UK. Retrieved July 12, 2015, from https://insidegovuk.blog.gov.uk/2015/06/17/the-new-normal-for-gov-uk/

U.S. Census Bureau. (2015, February 17). *Quarterly Retail E-Commerce Sales.* Retrieved July 12, 2015, from http://www2.census.gov/retail/releases/historical/ecomm/14q4.pdf

Van DeGrove, J. (2010, December 8). Each Month 250 Million People Use Facebook Connect on the Web. Retrieved September 4, 2015, from http://mashable.com/2010/12/08/facebook-connect-stats/

Veitch, D., Flandrin, P., Abry, P., Riedi, R., & Baraniuk, R. (2002). The Multiscale Nature of Network Traffic: Discovery, Analysis, and Modelling. IEEE Signal Processing Magazine.

Welcome to GOV.UK. (2015, July 12). Retrieved July 12, 2015, from https://www.gov.uk/

Woods, D. D. (1993). Process-tracing methods for the study of cognition outside of the experimental psychology laboratory. In C. E. Zsambok, G. A. Klein, kl, R. Calderwood, & J. Orasanu (Eds.), (pp. 228–251). Norwood, NJ: Ablex Pub. Retrieved from http://psycnet.apa.org/psycinfo/1993-97634-013

Woods, D. D. (1995, January). Cognitive demands and activities in dynamic fault management: abductive reasoning and disturbance management. In *Human factors in alarm design* (pp. 63-92). Taylor & Francis, Inc..

Woods, D. D. (1995b). The alarm problem and directed attention in dynamic fault management. *Ergonomics*. http://doi.org/10.1080/00140139508925274

Woods, D. D., & Patterson, E. S. (2001). How unexpected events produce an escalation of cognitive and coordinative demands. *Stress Workload and Fatigue. Hillsdale, NJ: Erlbaum.*

Woods, D. D., & Hollnagel, E. (2006). Joint cognitive systems: Patterns in cognitive systems engineering. CRC Press.

Woods, D. D., Patterson, E. S., & Roth, E. M. (2002). Can we ever escape from data overload? A cognitive systems diagnosis. *Cognition, Technology & Work*, *4*(1), 22–36.

Wright, P., & Monk, A. F. (1989). Evaluation for design. *People and Computers V*, 345–358.

Yin, R. K. (2013). Case Study Research. SAGE Publications.