Adaptive Tactics in Software Operations: How People Manage Complexity by Asking for Help

Michael Wettick | LUND UNIVERSITY

Adaptive Tactics in Software Operations: How People Manage Complexity by Asking for Help

Michael Wettick

Lund 2024

Adaptive Tactics in Software Operations: How People Manage Complexity by Asking for Help

Michael Wettick

Number of pages: 61
Illustrations: 13

Keywords
Recruitment, Software Operations, Anomaly Response, Incident Response, Escalation, Adaptive Capacity, Decision Making, Tacit Knowledge, Complexity

Abstract
Society relies on the continued operation of software systems, from healthcare to social media platforms. These systems can be engineered to be highly reliable, but complexity assures surprise. People have the ability to manage this complexity, filling gaps in design by recognizing, diagnosing and correcting anomalies. Rarely can surprises be managed by individuals alone. When engaged in anomaly response, people ask for help, recruiting others into the response.

This study explores recruitment – what criteria influence people to recruit additional resources, who do they recruit, why do they recruit them, and what methods do they use for recruitment? A case study approach was used, focusing on instances of recruitment across four cases of anomaly response in software operations. Data were collected in the form of chat transcripts, and records from monitoring and alerting tools and code repositories. These data were used to inform semi-structured interviews and cued recall with participants to elicit perspectives related to decision-making around specific instances of recruitment.

In attempting to maintain operations, individuals sought help for a variety of reasons: diagnosis, repair, coordination, cross-checking, information gathering and approval. Recruitment occurred across a range of roles, and requests for help occurred throughout the response. People also employed various strategies in their attempts to recruit others, such as communicating urgency by sharing a “scary” graph and leveraging the organizational structure to identify who might be available to help. In all cases, restoring system operations required recruiting additional people into the response, and this recruitment was enabled by the accumulated experiences and relationships of the recruiters.

Riskhantering och samhällssäkerhet
Lunds tekniska högskola
Lunds universitet
Box 118
221 00 Lund

http://www.risk.lth.se

Telefon: 046 - 222 73 60

Division of Risk Management and Societal Safety
Faculty of Engineering
Lund University
P.O. Box 118
SE-221 00 Lund
Sweden

http://www.risk.lth.se

Telephone: +46 46 222 73 60

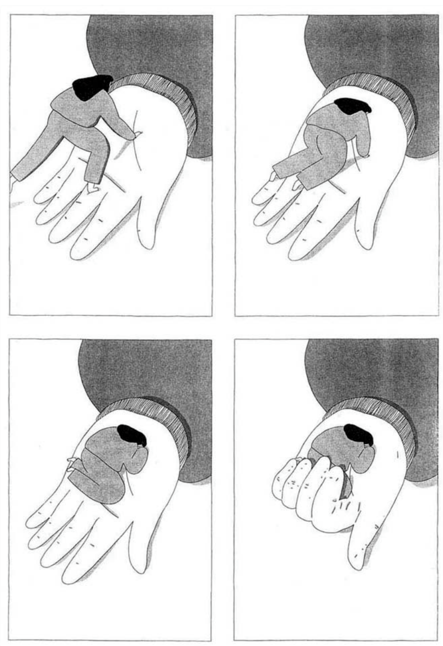
Acknowledgments

First, thank you to my remarkable wife and partner, Katie. You inspire me, you challenge me, and every day I feel your love and support. And yes, you are the brains in this relationship.

John Allspaw, I would not be in this program if not for you. I was contentedly living in a much simpler world before you ruined it with your talks and frustratingly brain-twisty perspectives and insights. Though we did not know each other at the time, you altered the course of my life in a significant and positive way. It has been incredibly fulfilling to work with you on this research and to have your time and support as my supervisor.

In the last few years, I have been influenced and shaped by so many wonderful people. Dr. Richard Cook, Professor Johan Bergström, Dr. Anthony Smoker, "Commander" Katie Metzroth, Nora Jones, Dr. Laura Maguire, Colette Alexander, Chad Todd, and Alex Elman— thank you. I want to be like you when I grow up.

Finally, thank you Kathy and Kevin for always seeing the best in me.



Marsh, Amy V. (n.d.). Accept Help. https://www.yayvm.com/

Contents

List of Tables and Figures

Introduction

The availability of performant software is essential to the safe operation of critical infrastructure such as healthcare, emergency services, energy production, and transportation systems. Recognition of the hazards of software in high-consequence industries is not new (Leveson, 1986); however, the increasing digitalization of society has made more ambiguous the boundaries of what is safety-critical. Temporary unavailability of social media platforms can prevent access to emergency services and destabilize local economies (Jilani, 2021). We are increasingly running the critical infrastructure of society on services previously viewed as conveniences (Allspaw, 2022).

Software systems are complex. Digital services consist of many interdependent components distributed geographically and across organizations. The interrelated nature of components creates dependencies between the individuals, teams, and organizations that support them. These dependencies create situations where cause and effect are not apparent, the implications of decisions can be unpredictable, and unforeseen emergent behaviors threaten the system's functioning (Senge, 2006). A small change that degrades one software service can start a chain reaction in tightly coupled dependencies, ultimately bringing down large swaths of the internet (Hern, 2021).

Complex systems can be fragile. Perrow (1999) argues that catastrophe is inevitable in tightly coupled complex systems as situations arise from the interaction of components, unforeseeable by any designer. Surprise is inevitable – in software, it's expected – "things that have never happened before happen all the time" (Sagan, 1993, p. 12). Digital services can be highly reliable and protected against many failures (Beyer et al., 2016). However, changes that increase reliability also create additional complexity, making the system further susceptible to fundamental surprise (Lanir, 1986) and sudden collapse (Ormerod, 2007; Alderson & Doyle, 2010).

Systems can adapt. The warnings of Perrow (1999) are not a certainty. Woods & Branlat (2011) describe observable patterns in how systems fail to adapt to surprise:

- *Decompensation*: exhausting capacity to adapt as challenges cascade
- *Working at cross-purposes*: behavior that is locally adaptive, but globally maladaptive

- *Getting stuck in outdated behaviors*: the world changes, but the system remains stuck in what were previously adaptive strategies

Systems can then develop capabilities to monitor for these adaptive traps and strategies to manage them. Cook (2006) describes techniques used to preemptively identify bumpable patients that can be transferred from the ICU to generate additional capacity for new patients. Patterson & Woods (2001) observe how space shuttle mission control is designed to rapidly call in additional expertise and get them informed on the state of the system. In software, organizations create and experiment with formal processes for the recruitment of resources as capacity demands arise (Cook & Long, 2021).

People are the adaptable part of a system. Ashby's (1958) law of requisite variety states that the controller of a given system must have at least as much variety as the system to control. While it is impossible to design for all possible states of the system, things can go right because people can overcome shortcomings in the design, at times going against training and procedures, adapting their performance to meet current demands (Hollnagel, 2012). People have the requisite variety to manage complexity, and the tactics used to manage this complexity are observable and cultivatable.

One way people adapt is by asking for help. As situations move from routine to exceptional, people become concerned that events may take an unexpected and undesirable direction that may require action. During this process of problem detection (Klein et al., 2005), individuals may engage more resources and types of expertise. Woods & Wreathall (2016) distinguish between two forms of this adaptive behavior in a system:

- *First order adaptive behavior*: Following a pre-planned response, using rules, checklists, and pre-allocated resources to meet the foreseen demand.
- *Second order adaptive behavior*: The capacity of the system to engage more resources to meet unusual demands.

To what degree a system can recruit the necessary resources to meet unusual demands influences whether a system is brittle – moving quickly towards the failure point and collapse – or resilient.

What adaptive tactics (Anders et al., 2006) do people use in recruiting resources to meet unusual demands? The ability of people, or teams, to effectively bring the necessary skills, experience, and authority to bear for a given situation influences whether a system can stretch, or risks saturation and failure. Asking for help is an adaptive tactic to manage challenges and uncertainty in complex systems.

This research will focus on recruitment activities (asking for help) by individuals and teams involved in software anomaly response:

- What criteria influence engineers to recruit additional resources?
- *Who* do they recruit, *why* do they recruit them, and what methods do they use for recruitment?

Theoretical Background

The questions motivating this thesis are posed to better understand the above-the-line (Cook, 2020) cognitive work of anomaly response when operating complex software systems. While Allspaw (2015) noted the dearth of research investigating anomaly response in the software domain, research by Allspaw (2015), Grayson (2018), and Maguire (2020) have since built a foundation that this research is inspired by and aims to contribute to.

As organizations operate complex software systems, anomalies inevitably threaten the system's operation. Operators then engage in anomaly response (Woods & Hollnagel, 2006) to maintain a desired or expected state, processes which include: recognizing what is anomalous, diagnosing what is producing anomalies, analyzing the implication of anomalies, and modifying plans and re-prioritizing goals to maintain control.

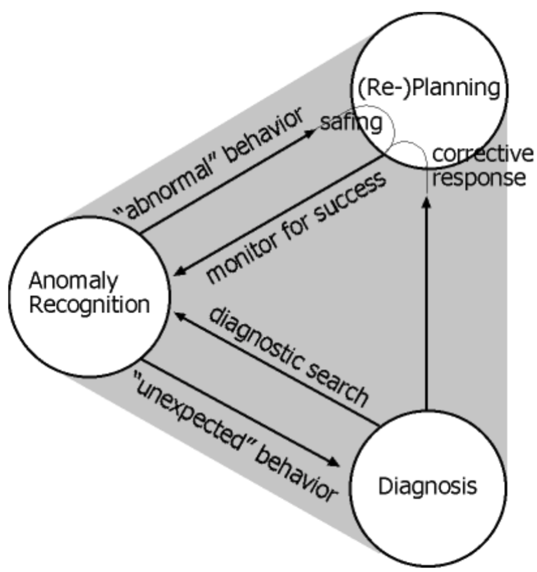


Figure 1. The intermingled lines of reasoning in anomaly response (Woods, 1994)

Observations of anomaly response have noted a recurring pattern of escalation, a relationship where the greater the trouble in the underlying process, the greater the information processing activities required to cope with the situation (Woods & Patterson, 2001). As a situation escalates and workload demands increase, operators have four coping strategies (Woods & Hollnagel, 2006, p. 126):

1. Shed load
2. Do all components but do each less thoroughly
3. Shift work in time to lower workload periods
4. Recruit more resources

Additionally, as the problem cascades, more knowledge or different skills may be needed, which may reside in different people or parts of the operational system. Recruitment of resources is then both a coping strategy for managing workload demands and, at times, necessary to maintain control of the system.

Operating software systems is inherently collaborative due to the interdependent structure of the system and characteristics of the work. Work in software, or with any complex technology, is characterized by ambiguity, uncertainty, and having to take actions on the system where policies and procedures to define the task often do not exist (Weick, 1990; Griffin et al., 2007). Given this uncertainty, particularly when the current state of the system conflicts with expectations, individuals will attempt to make sense of the situation, asking questions such as "why did it do this?", "what is it doing now?", and "what will it do next?" (Woods & Sarter, 2000). As part of this sensemaking process, people may turn to others for help to clarify what is happening in the system, and what to do next (Weick et al., 2005; Maitlis, 2005).

Anomaly response literature makes frequent references to these recruitment activities having occurred during response:

- "Additional personnel were called in and integrated with others to help with the new workload demands" (Woods & Hollnagel, 2006, p. 70)
- "The team expanded to include a large number of agents in different places" (Woods & Hollnagel, 2006, p. 70)

- "The lack of expected signals hinted at a problem on a different level, forcing them to broaden the scope of their investigation and recruit other roles and expertise" (Grayson, 2018, p. 70)

These observations in the literature establish recruitment as a critical element of anomaly response. However, references to recruitment are primarily only observations that it occurs and do not explore in detail the specific actions and methods of recruitment or cognitive work involved.

Maguire (2020) presents the most detailed observations on recruitment and how it influences or is influenced by other elements of choreography during anomaly response. Maguire (2020) identifies overhead costs associated with recruitment, such as:

- Monitoring current capacity relative to changing demands to recognize when additional help is needed
- Identifying the skills required
- Identifying who is available
- Determining how to contact them

Maguire (2020) also describes characteristics of recruitment, observing that it can be effortful or lightweight depending on the methods used, can happen abruptly or ad-hoc, and differs depending on if recruitment is happening within a team, across team boundaries, or across organizational boundaries.

How do people decide to recruit? Klein (1993) finds that people make decisions by reacting on the basis of prior experience, particularly when under time pressure. When giving accounts of their decision making during emergency events, fireground commanders argue they do not "make choices" or "consider alternatives" (Klein, 1993, p. 2). There are decision points, but the advantages and disadvantages of various options are not weighed in the moment. Rather, these experienced decision-makers identify an acceptable course of action as the first one they consider. This process of recognition-primed decision-making involves the recognition of a situation as familiar and, with that, the recognition of familiar goals, cues, and typical reactions (Klein, 2008).

More broadly, research in psychology and organizational behavior has explored help-seeking behavior for individuals at work. Help-seeking being defined as the act of asking others for assistance, information, advice, or support (Lee, 1997). Hofmann et al. (2009) find nurses are more likely to seek help from peers they perceive as experts, provided they also perceive them as accessible and trustworthy. van der Rijt et al. (2013) further describe these relational aspects of help-seeking, noting that individuals are likely to seek help up the organizational hierarchy and reiterating the importance of trust, accessibility, and expertise. Jansen et al. (2021) investigate the willingness to ask for help and find this to be influenced by a perceived need to come across to others as certain, decisive, and independent. Furthermore, there are psychological or cultural factors that may discourage help-seeking, such as adversity triggering anxiety and impulses to flee or hide, which undermine coordination (Barton & Kahn, 2019), and in-group/out-group dynamics that can fragment an organization and create barriers to giving and receiving support (Kahn et al., 2018).

Commonly, literature related to anomaly response in the software domain is focused on reducing uncertainty through technical design and improvements in information processing capabilities. However, uncertainty is not as much a function of information processing capacity as it is the ability of the designer to predict and identify all conceivable states of the system (Shattuck & Woods, 1997). There will always be uncertainty, and operators will always confront unforeseen situations and incomplete information. It is important to understand and support the tactics people use to cope as situations move from a thought-of to an unthought-of state, of which the mobilization of additional resources is a primary tactic (Cuvelier & Falzon, 2017).

This thesis aims to better understand help-seeking behavior as it occurs as recruitment during anomaly response by analyzing specific instances of recruitment across a small selection of cases. At a high level, the intent is to provide a description of how help-seeking is used by individuals to navigate uncertainty in complex systems. Specifically, what influences individuals to seek help from others, what methods are employed to seek this help, and what makes this easy or difficult? An understanding of the cognitive work of recruitment and the challenges individuals face to recruit may inform the design of more efficient and safer work systems.

Software Incident Response

A brief, partial summarization of processes and tools people use when responding to software incidents is provided here as context which may be helpful to understand the cases in this study.

Software development and operations, and the people doing the work, exist between constant pressure for change from one side and demands for stability on the other. Organizations such as the one studied in this research serve a global audience, with users expecting near constant 24/7 operation.

When issues arise, engineering teams prefer to become aware of these issues through systems set up to constantly monitor the system and detect anomalies. Often these monitoring systems are tied to alerting systems, which will email, text, call or otherwise notify an individual who is "on-call" for whichever component in the system is experiencing a disruption. This on-call person then becomes responsible for diagnosing and resolving the issue, escalating to someone else, or initiating whatever process is in place for incident response at their organization. Issues may also occur in parts of the system that are unmonitored, or may be caught by monitoring but lost within a noisy alerting system, and ultimately be detected and reported by a human.

Maguire (2020) describes four common models of how people coordinate for incident response within software teams (though significant variations exist from one organization to the next):

- Incident Command model - a collection of formal and ad hoc groupings work under a central (and usually singular) leader
- One At A Time model - ownership for the problem is assigned to discrete groups responsible for different aspects of service delivery and passed along as the problem is found to reside somewhere else
- Escalation model - the problem gets escalated through different levels of responders
- All Hands on Deck - recruiting all available resources when the severity of the incident was deemed to be high

When multiple people are collaborating to resolve a software incident, tools are used to assist in this coordination. For organizations using chat platforms to communicate internally, these platforms often become the primary tool for communicating during incident response as well.

The four cases investigated in this research include a combination of incident coordination using alert tools, chat platforms and video calls.

Research Methods

Overview

This research adopts a qualitative research methodology, the central phenomenon (Creswell & Baez, 2020) being the act of recruitment during anomaly response. The research was conducted from a relativist ontological perspective, acknowledging that reality is a subjective and socially constructed experience (Crotty, 1998). The research pursues to report the perspectives of participants and acknowledges that the research results are a co-construction between the participants and the researcher based on our personal experiences and interactions (Creswell & Creswell, 2017).

In the introduction to this research, it was established that one observed coping strategy for managing the increasing workload of anomaly response is to recruit additional resources. This research explores the "how" and "why" of recruitment. Why do individuals recruit who they do, when they do, and how do they do it? What tools, processes, and methods do they use to recruit?

For qualitative research, a case study approach is appropriate when seeking to explore "how" and "why" questions – especially those where the researcher has little or no control over behavioral events (Yin, 2018). For this research, four cases were selected, all involving instances of software anomaly response within the same organization. The intent of selecting multiple cases was to study a variety of instances of recruitment and to provide an opportunity to contrast recruitment across cases. Data was collected for each case, including conducting interviews with participants.

One intent of the research is to better understand the cognitive processes involved in the decision to ask for help and actions taken to recruit others. As such, relying solely on verbal reports from semi-structured interviews is insufficient. As reviewed by Nisbett & Wilson (1977), evidence suggests individuals have little or no direct access to their own higher-order cognitive processes. When retrospectively recalling the factors influencing a particular action, individuals construct an explanation based on causal theories and judgments about what now seems plausible for the given situation, which may or may not be what occurred in the moment. Verbal reports are not entirely unreliable, such as when people are instructed to think aloud while performing a task (Bainbridge, 1979; Ericsson & Simon, 1980); however, the present research was conducted solely after the events of interest. It would have been preferable to conduct direct observation, but not practical due to the sporadic nature of software anomalies and the researcher's ability to be present for and have access to these events.

To support participants in reflecting on their experience of recruiting others during anomaly response, this research utilizes process-tracing methods inspired by behavioral protocol analysis (Woods, 1993). The objective of process tracing is to "lay out the problem solving episode from the point of view of the people in the problem", including identifying mismatches between a participant's perception of a situation that occurred when compared to evidence from the event (Woods, 1993, p. 236). This approach makes use of data from the events of interest to cue recall during semi-structured interviews and later correlate evidence from the event with what is explained in verbal reports. Woods (1993, p. 235) outlines some types of data that may be relevant for analysis:

- direct observation of participant behavior
- traces of data acquisition sequences
- traces of actions taken on the underlying process
- records of the dynamic behavior of critical process variables
- records of verbal communication among team members or via formal communication media
- verbal reports made following the performance
- commentaries on their behavior made by other domain knowledgeable observers

These various lines of evidence are then cross-referenced to establish a trace of participant behavior and cognitive activities, ultimately producing "a record of participant data acquisition, situation assessment, knowledge activation, expectations, intentions, and actions as the case unfolds over time" (Woods, 1993, p. 235).

A high-level overview of the process-tracing methodology used in this research – adapted from a similar figure in Allspaw (2015) – is presented in Figure 2.

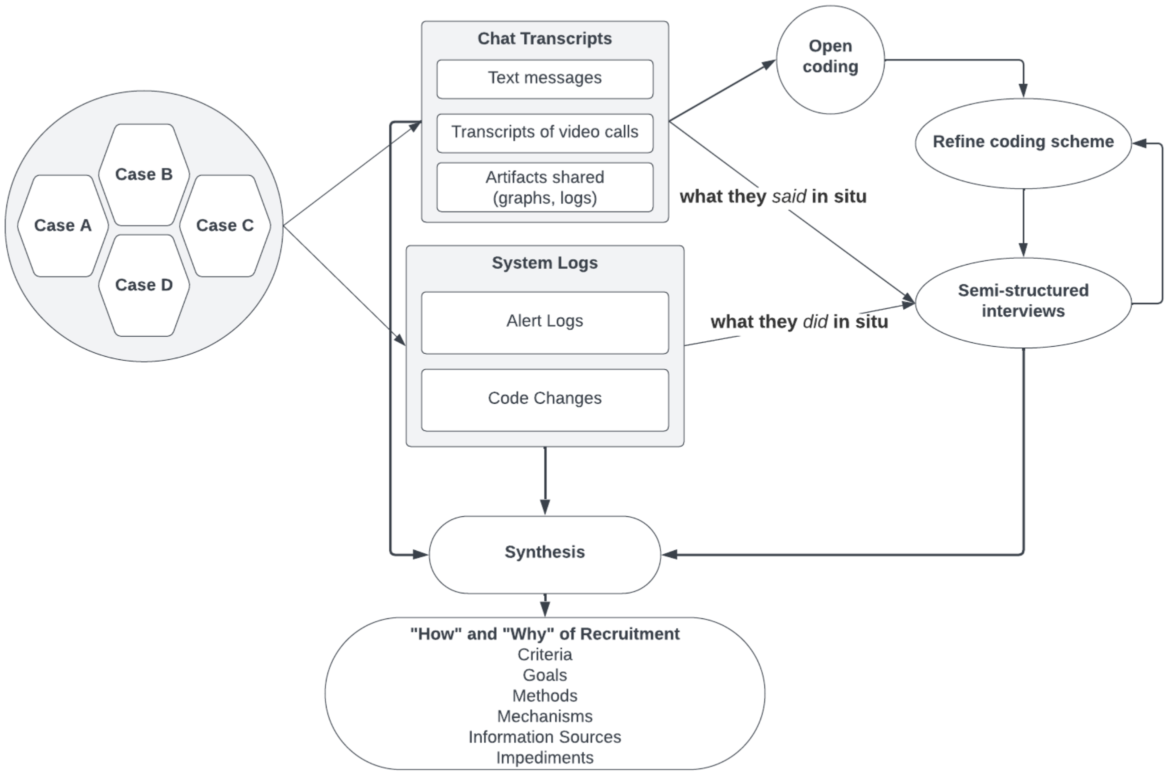


Figure 2. High-level overview of the transformation of behavioral and verbal protocols into findings

Participants and Data

Participants for this research were recruited from a large multinational corporation that operates a web platform as a primary line of business. A selection of four cases were identified amongst the regularly occurring operational incidents within this organization (Figure 3). Cases were identified based on criteria such as the duration of the event, the number of people involved, the number of teams or outside organizations involved, and the occurrence of recruitment actions during the event. As potential cases were identified, data from the event was reviewed to assess the relevance of the event to the phenomenon of interest. Once a case had been selected for further analysis, participants were contacted, and interviews were scheduled.

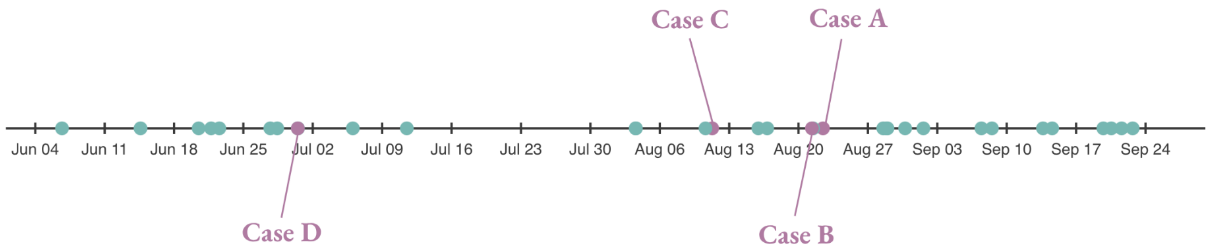


Figure 3. The selected cases from the stream of events

Given that the research is focused on anomaly response within globally distributed software teams, there was a substantial amount and variety of digital artifacts available for collection, such as:

- Records of communication by team members in chat logs and video/audio recordings
- Logs of system data that participants were viewing in monitoring systems during the event
- Alerts sent by both manually by individuals and from automated processes
- Code changes made to mitigate or resolve the issue

Case Selection

To start the process of case selection, the researcher was given access to Jeli (2024), an incident management platform, which contained data for over one hundred cases from that organization over the preceding three years. This repository contained data such as chat transcripts, a record of who participated in the event, and context about each participant, including their role, tenure, and geographic location (Figure 4).

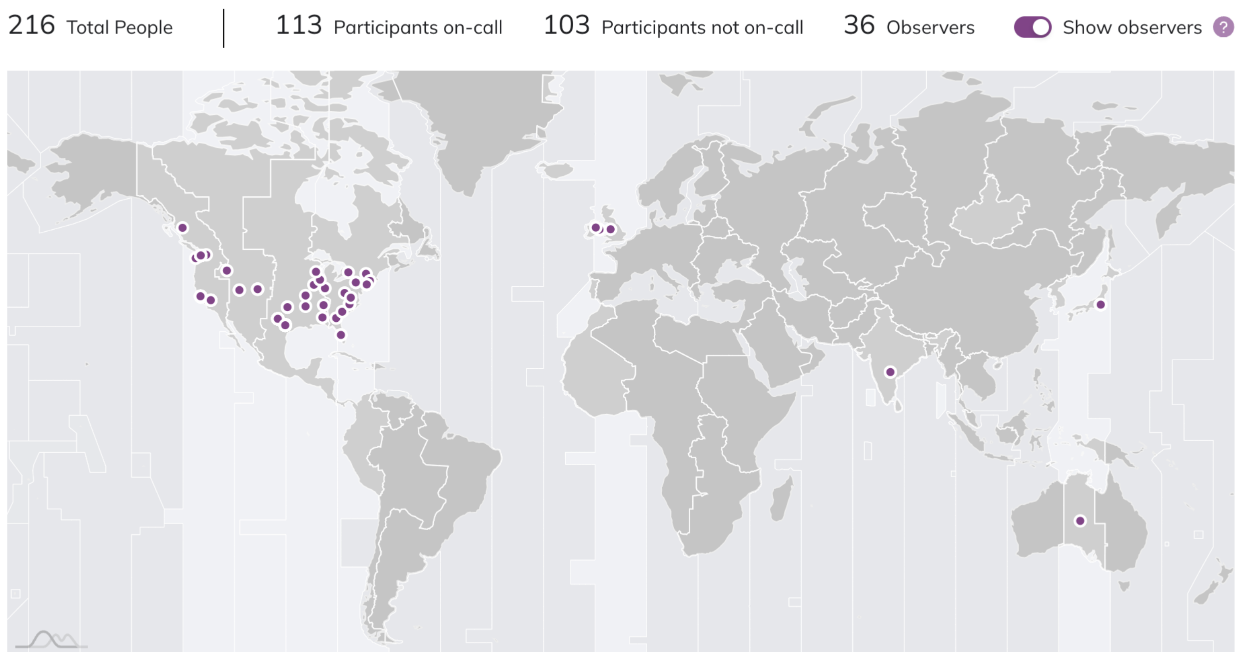


Figure 4. Example participant overview in Jeli

From this repository, twenty-seven cases were reviewed to assess the potential relevance to this research. Of the twenty-seven cases, eight were identified as potential cases for further data collection and participant interviews. All eight cases initially selected contained instances of individuals recruiting others to the response. These eight cases were then analyzed to create an initial coding scheme that differentiates between the instances of recruitment. These eight cases were further refined to the four cases presented here in the research.

Event Data Collection

The following data were collected:

- Case A
  - 4 chat channel transcripts
  - 1 interview with an incident responder
  - Records from alerting tool, code repository, internal company wiki and monitoring system
- Case B
  - 2 chat channel transcripts
  - 1 interview with an incident responder
  - Records from alerting tool and monitoring system

- **Case C**
    - 2 chat channel transcripts
    - 1 interview with incident responder
    - Records from alerting tool
- **Case D**
    - 3 chat channel transcripts
    - 1 interview with an incident responder
    - 1 recording of the incident video call
    - Written documentation of the event created by responders during the incident

The majority of data for the cases is communication data in the form of text chat logs. An example is seen in Figure 5 – the utterances collected for Case A across four communication channels. See Table 1 for a more detailed breakdown of channels, participants and utterances by case.

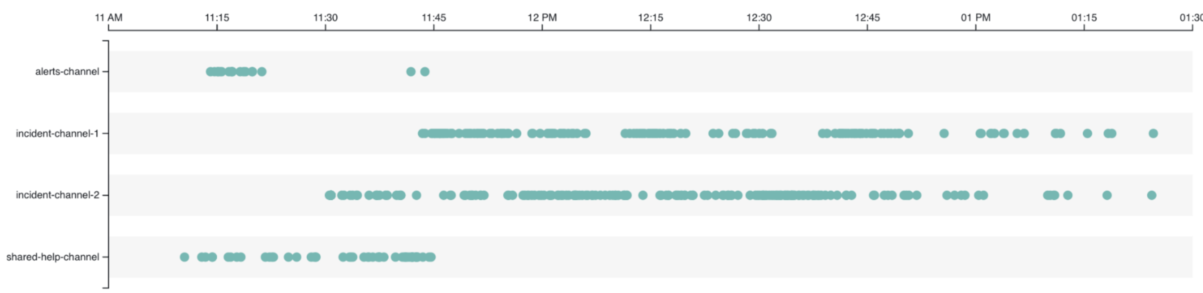


Figure 5. Utterances over time by chat channel for Case A

In addition to chat logs, data for Case D included a 4-hour 50-minute recording of a video call that occurred during the event. A software service was used to perform the first pass of transcription for this video file, converting the audio to a text transcript. The researcher then listened through the recording and made modifications to the software-generated transcript where appropriate.

The chat data collected are not assumed to be the complete collection of all chat data by responders during these cases. The channels collected and analyzed were confirmed by interview participants to be the official channels for communication regarding these events.

Table 1. Overview of corpus cases

Case	Duration	Participants	Channels	Utterances
A	121 minutes	61	4	427
B	107 minutes	11	2	113
C	54 minutes	16	2	283
D	553 minutes	180	4	1209

Artifacts from monitoring systems and alerting tooling were also collected (Figure 6) as additional context when analyzing chat transcripts, and for corroboration with statements made during interviews.

Figure 6. A monitoring system graph used as context in recruitment in Case A

Instances of recruitment were found when collecting and analyzing data from alert tools that were not visible solely from the chat transcripts (Figure 7).

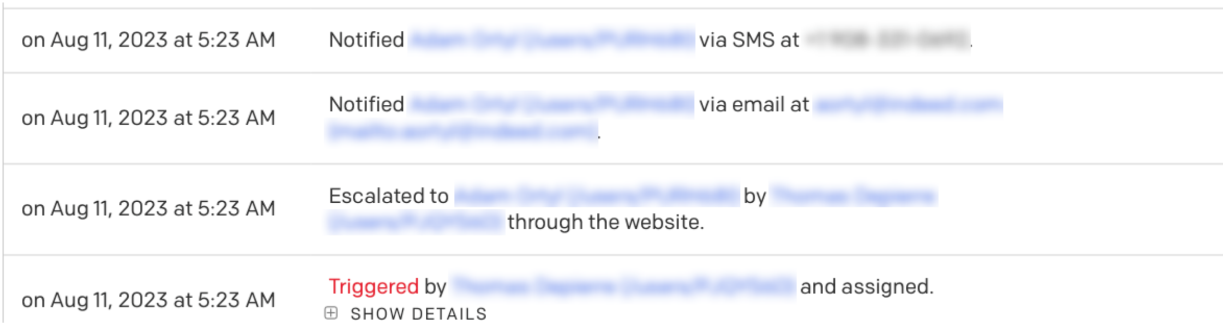


Figure 7. Recruitment via alerting tooling from Case C

There was additional data that may have been relevant to the research that was not available, such as direct message communication privately sent between individuals involved in the events. Efforts were made during interviews to ascertain whether private communication occurred during the event and, if so, the content and recipients of those messages.

Interviews

Participants from each case were selected for semi-structured interviews inspired by the Critical Decision Method (CDM) (Crandall et al., 2006). The goal of the CDM interviews was to elicit participant's perspectives about decision-making, planning, and sensemaking as it relates to the event being discussed, with a specific focus on instances of recruitment.

For each case, a participant was selected who had been involved throughout most of the case and had recruited one or more other individuals into the response. Each case had multiple options to interview, and participants with more examples of recruitment who had been involved throughout more of the response were preferred. For example, Figure 8 shows all 10 individuals who had recruited someone into the response for Case A. Responder R1 was selected for interviews based on them having the most instances of recruitment visible in the chat logs that spanned the length of the event. Interviews were limited to one person per case to contain the scope of the research.

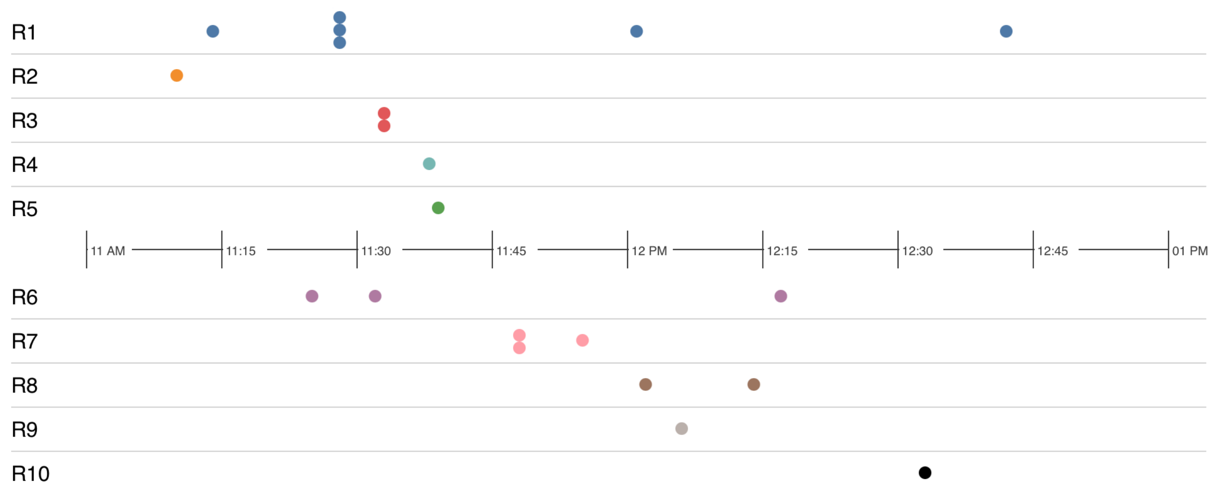


Figure 8. Recruiters (R1-R10) and their instances of recruitment during Case A

CDM interviews are typically conducted in four phases:

1. Incident Identification
2. Timeline Verification
3. Progressive Deepening
4. "What If" Queries

Each phase uses different techniques to help the participants recall events in greater detail. Probe questions are used to explore how factors such as cues, information, goals, priorities, and experience influenced decisions to recruit additional resources into the incident.

Incident identification was not necessary for this research, as cases had been selected prior. Additionally, having already identified a case, the researcher came into the interview with a timeline constructed based on the utterances from the interview participant and others from the event transcripts. Because of this, the researcher conducted an abbreviated form of timeline verification, asking the participant to share their memory of how the event unfolded first, before going deeper into cued recall.

During the interviews, CDM “deepening probes” were used when asking participants about specific utterances they made in the chat transcripts. Crandall et al. (2006) provide examples of deepening probes that inspired interviewing questions for this research (p. 79):

- What information did you use in making this decision?
- How and where did you get this information, and from whom?
- What were your specific goals and objectives at the time?
- What other courses of action were considered or were available to you?
- What let you know that this was the right thing to do at this point in the incident?
- How much time pressure was involved in making this decision?

"What If" queries were also used, primarily to elicit expert-novice contrasts.

FROM INTERVIEW

Researcher: "So if a new engineer were dropped into your position, seeing those alerts come out, where do you think they would struggle?"

Engineer: "Well, I think there's the knowledge of knowing who to ping. That's one. Knowing when it's an API team, who to get involved. I think sometimes there's the differentiation of how important certain alerts are. We have different alerts and they aren't necessarily marked as this one's critical or this one's not. And sometimes it can be hard to determine this is a terrifying alert versus this is a thing that happens and isn't great, but it happens. And then I think there's just a personality thing where you have to get comfortable bothering people when you're dealing with an event. It feels uncomfortable to be paging people, but it's far worse to do nothing or be hesitant because if you're hesitant, then that can really be bad for you long term. If you don't address something that costs millions of dollars, it's a lot worse to be like, well, I was scared about messaging - I didn't want to bother them when it's like you could have said you saved the company millions of dollars, but you were scared to do that. You know what I mean?"

Interviews were recorded, and transcription was conducted in a two-part process, with a first pass by software transcription, followed by a pass of manual adjustment by the researcher.

Coding

An initial coding scheme was created by analyzing chat transcript data during case selection. This coding scheme was a list of observations as the researcher perused the transcripts, for example:

- The actor(s) being recruited may be a person, team or an outside entity
- Requests for help may be named – requesting a specific person/team/entity, or unnamed (needing help from someone with experience with a particular tool, skill or technique)
- Requests for help are communicated with words ("Could you help me with X?") and/or artifacts (system logs, graphs) intended to convey a request for help
- Requests are made for a variety of purposes: help diagnosing an issue, handing off responsibilities, help coordinating work

Interviews with participants provided more insight into the specific goals of responders, the criteria that motivated them to request help, and why they asked who they did for help. Following the first interview (Case A), this additional insight was added to the list of observations from chat transcripts, creating a first coding scheme within the qualitative data analysis software NVivo (2024).

As interviews were conducted and coded, some codes were further broken down into sub-codes, and others were combined. Iteration on the coding schema occurred continuously throughout the research, with data being re-coded in multiple passes.

Table 2 presents examples of top-level codes, codes and example utterances:

Table 2. A subset of codes, taken from the final coding schema

Top Level	Code	Description	Example(s)
Recruiting	Direct	Recruiting a specific person, team, or organization by name	*- I paged the API team to join the incident* *- I asked Eva who could help me here*
	Indirect	Recruiting others without expressed intent for any specific person, team or organization	*- I posted a thread in a shared channel with observations of what I was seeing*
Recruitment goals	Seeking additional capacity	Recruiting for help with tasks (diagnosis, repair, coordination, communication), or to help recruit others	*- I need someone to help look at this and fix it* *- They're supposed to help facilitate the whole incident* *- They can find the person I need to be talking to on their team*
	Seeking support	Recruiting for help cross-checking decisions, providing information	*- I'm wondering if we need to send out comms* *- I'm not completely comfortable saying this is how we should do it*
	Seeking authority or approval	Recruiting for approval to take some action	*- I need somebody to make a call*

For the complete, final coding scheme, see Appendix A.

Ultimately, chat logs and incident call audio transcripts from the four cases were only coded for recruitment (direct/indirect). Interview transcripts with participants were coded for criteria, recruitment goals, sources of information, and recruitment difficulties.

Corpus Description

Provided here are summaries of each case to provide a sense of what happened, who was involved and the perceived urgency or severity of the issue. These descriptions are not intended to be a thorough representation of the event.

Case A: Scary graph

Around 11:15 AM, multiple engineers started noticing anomalous traffic data for their services. One engineer, upon examining monitoring data, called in his team lead for help investigating. Another engineer posted into a chat channel meant for discussing potential issues: "..is anything going on right now? My service is seeing a complete drop in requests...". This drop in traffic was perceived by engineers to represent a significant impact to the availability of the site. In the hours preceding this event, monthly maintenance to update some components in the system had caused some tests to fail, generating alerts. Multiple additional engineers joined the response to provide this context, suggesting this maintenance was also responsible for the drop in traffic.

Investigation continued in the help channel until 11:43 AM ET, when a manager asked if the organization's formal incident response process had been initiated. A new channel for responding to this issue was created. Shortly before the creation of this new channel, at 11:30 AM ET, a different incident response channel was created to investigate different symptoms within the system related to the same issue. Different sets of responders investigated the issue in both channels in parallel. Responders realized during the event that the two channels were investigating the same underlying issue and directed all communication to one of the two channels.

Responders were able to determine that a change to a framework shared by many applications had been made at 10:55 AM ET, correlating with the beginning of the impact as indicated in the monitoring system. This shared framework handles receiving and routing requests to other applications, and the change had been made to this routing component, causing requests to be deemed unauthenticated and not passed on to the target application.

This routing authentication change was reverted in the system, and normal operations resumed.

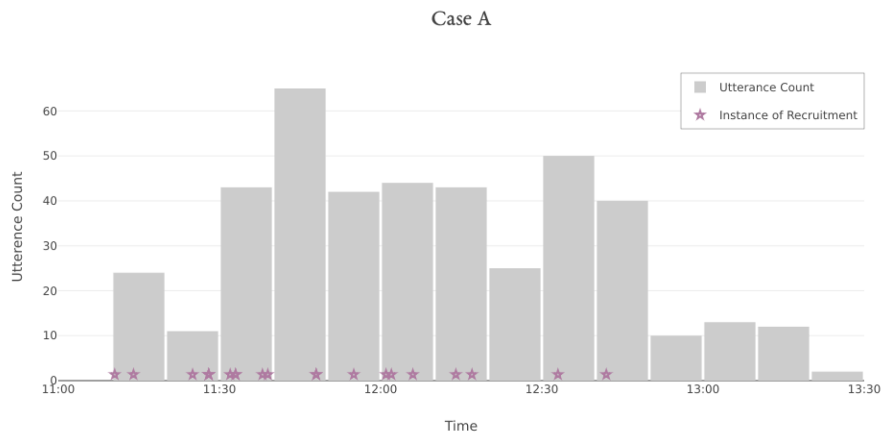


Figure 9. Distribution of utterances during Case A

Case B: Turn it off and on again

An on-call engineer received an automated alert at 7:45 AM ET titled "...processing lag is over 30.0 minutes in prod", with some context that provided a link to documentation for the service. Within the documentation, the engineer read that this processing lag may be user-impacting, preventing users from seeing up-to-date data within the application.

The engineer investigated the issue for 39 minutes before escalating the alert to another engineer for assistance at 8:24 AM ET. The engineer also created a new communication channel for this event – which is policy and common practice at the organization – and recruited a manager at 8:30 AM ET to help handle communication.

The engineer was able to determine a service that may be responsible for the issue, and attempted to recruit someone from the team that owns that service to the response. There was confusion about which team to recruit due to inconsistency between the team name, service name, and the name used within the alerting tool.

The first person the engineer tried to recruit joined the channel and shared that a similar issue with this service had happened recently. The responders decided to restart the service assumed to be causing the issue, and operations quickly returned to normal.

After operations had been restored, an engineer from the service-owning team joined the event channel and shared context on what happened with the service to cause this issue:

> "Restarting … was the correct workaround. The issue is that the Java regex matcher by default is uninterruptible, and so if a regex hangs or takes a very long time then it can hijack one of the worker threads. If that happens enough times, all the worker threads can get stuck and then [the pipeline] stalls 😱. There is a fix ready to go that makes the regex interruptible so we can actually cancel the thread, but it wasn't ready until late Thursday and I don't have merge permissions :facepalm: As soon as I get permissions or get somebody to merge it for me, we'll get it out."

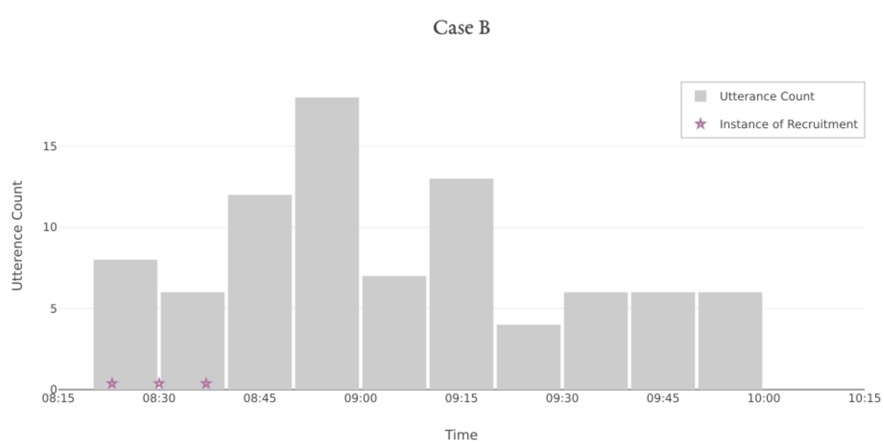


Figure 10. Distribution of utterances during Case B

Case C: Heisenbug

A sister company of the organization was experiencing increased error rates across multiple APIs and reported this by alerting an on-call engineer and posting into a shared channel meant for coordinating between the companies.

The on-call engineer began investigating, and the message into the shared channel was noticed by additional individual contributors and a manager, who then joined the response.

Both the on-call engineer and a manager in the response attempted to recruit an additional responder that had more in-depth knowledge of the failing APIs. As responders continued investigating and attempting to recruit others, the issue began to resolve itself without intervention by responders.

It was determined later that a vendor of the organization that provides a dependency for these services was experiencing an issue that caused the increased failure rates in their services.

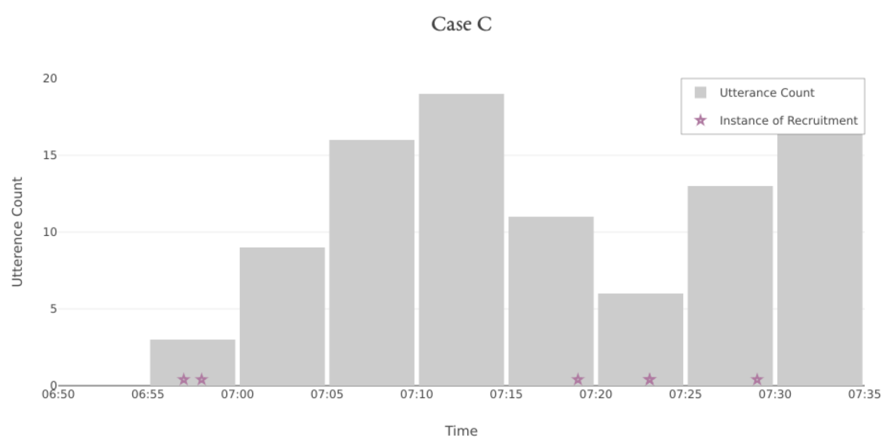


Figure 11. Distribution of utterances during Case C

Case D: The one where everything breaks

An on-call engineer posted in a large shared channel "Bunch of pages went off at once. Taking a look 1 by 1". After a few minutes of receiving an increasing number of pages, the engineer initiated the formal incident coordination process, creating a new communication channel for coordinating a response to this event.

Multiple engineers and managers across multiple teams quickly joined this channel, reporting issues with different services and rapidly falling metrics for company-wide key performance indicators. Many responders began notifying additional people, both requesting help and to spread awareness that a significant issue was occurring. Shortly after the event channel was created, a video call was created for responders to discuss the multiple lines of ongoing investigation.

An email was received from the organization's cloud provider notifying of an issue on-going at the cloud provider causing services to be unavailable. Given the breadth and severity of impact, responders began drafting communications to be sent to executives, notifying them of the situation.

Help was requested from the cloud provider, via both support ticket and chat messages. Options to mitigate the cloud provider outage and restore service were pursued on multiple fronts. Responders debated whether or not to display a message to users, notifying them that services were unavailable, and requested help from people in the product organization to advise.

The cloud provider gave an update that power had been lost at their facility, and that teams were working to restore it. Hours later, power was restored and the previously impacted services began to recover. Responders worked together on the video call and in different threads within the primary event communication channel to verify services had recovered, with some services requiring additional repair after service was restored at the cloud provider.

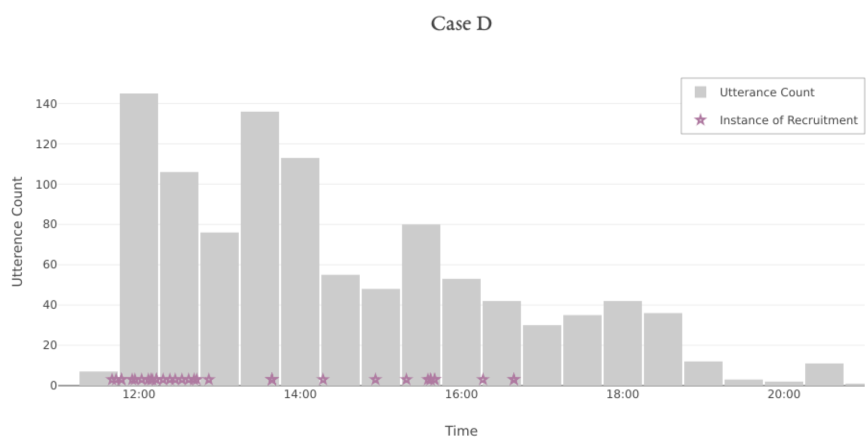


Figure 12. Distribution of utterances during Case D

Ethical Considerations

The ethical considerations regarding the participants in the research are under the guidance of Lund University's Research Ethics (Lund University, 2023). According to Lund University guidelines, this research did not require ethical approval. When partnering with the organization from which cases were selected for this research, approval was sought and granted by the necessary legal and compliance representatives. Informed consent agreements were distributed to, and signed, by all interview participants for this research (see Appendix C).

Results

The following is a description of the findings that resulted from the process of data collection and analysis described previously. Chat transcript data and information gathered from monitoring and alerting systems have been combined with participant interview responses across the cases, resulting in specific findings on recruitment across multiple dimensions: criteria that influenced the decision to recruit, goals of recruiting others, methods and mechanisms of recruitment, information used to inform who to recruit, and impediments to successfully recruiting others (Figure 13).

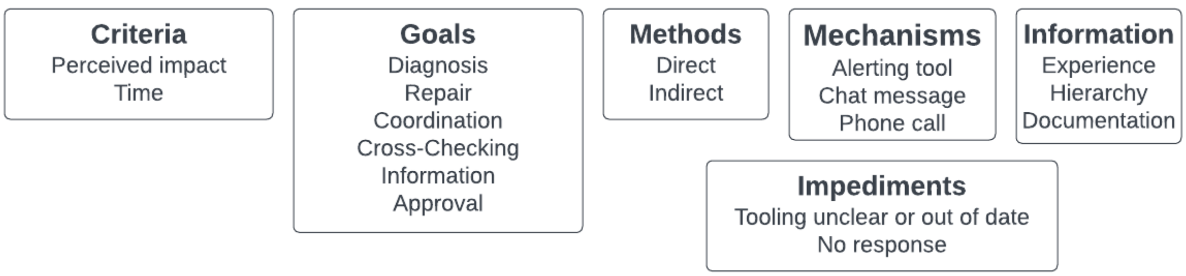


Figure 13. Illustration of the resulting groupings

Within the observed instances of recruitment, help was sought during response from both individual contributors ("hands-on" practitioners, referred to as ICs) and managers, seeking help from other ICs and managers. All instances of recruitment were initiated from individuals within the engineering organization, however, help was sought from people in both the engineering and product organizations.

Methods

All 4 cases involved a mix of both direct and indirect recruitment. Direct recruitment being instances where someone decided they need help, and went directly to a specific person or team by any mechanism (chat message, alert tooling, etc.).

```
[Case A] (incident-channel): "cc @<team> are you all in here?
Would at least one of you join the zoom?"

[Case B] (incident-channel): "Paging <team> on the <service> will
update with link shortly"
```

[Case C] (incident-channel) “@<engineer> is familiar with the ingress and failover, if we can get him on.”

[Case D] (incident-channel): “@<engineer> can you take a look at [application]? <link to monitoring dashboard>

[Case D] (incident-channel): “is someone from <team> here that can help join our call to troubleshoot [the service]?”

Instances of indirect recruitment include situations where someone decided they needed help, sought help, and did not request a specific person or team by name. For example, this engineer solicited help by posting to a chat channel with many members representing many roles and functions across the organization:

[Case A] (shared-help-channel): “I don't see any alerts, but is anything going on right now? My service is seeing a complete drop in requests”

A description of this channel is provided by one of the interview participants:

> “So we have a big slack channel that all the [developer first responders] are in and [site reliability engineers] and people high level, and when we get alerts in our team, if it looks severe enough, we'll escalate it into that channel.”

Three of the four cases in this research involved messages posted to this channel early in the response. In two of the four cases, this was followed by new responders joining the response and sharing findings in a response thread to the message. A similar pattern was observed in Case C, with a different channel:

[Case C] (shared-external-channel): “Hey team, starting a thread here for the elevated error rates we’re seeing for [feature].”

Indirect recruitment was at times unclear to the researcher, when solely looking at chat transcripts. For example, in Case B, the on-call engineer decided they needed help. They had started investigating the issue upon receiving an alert from the monitoring system. Within the alerting tooling, there was a predefined “escalation policy” that defined, specifically, who the next person to escalate to should be.

The engineer in this case escalated to the next person in the policy, and mentioned them by name in chat, but was not seeking to recruit help from them specifically. This act of deferring to a defined process when needing help is coded in this research as indirect recruitment.

Additionally, help is indirectly recruited into response by way of recruiting a particular function. For example, paging in whoever is currently on-call as the "incident facilitator" role, without specific intent for any one person to serve in this role.

> `[Case B] (incident-channel): "Created incident facilitator page here for comms <link to alert in paging tooling>"`

Indirect recruitment ("I need help for something and don't know who can help, or do not have a preference for who helps") was done using both chat and alerting tooling.

Mechanisms

Recruitment in Case B is an example of recruitment primarily using alerting tooling specifically intended to engage others into the response. This was supplemented by also notifying the recruit (called an 'at' mention) within chat, but the recruiter explained that the chat notification was to help the recruit find the right communication channel, and was not expected to be how the individual would be informed that their help was needed.

> "I find that to be the most effective way to just [use alerting tooling] and at tag people, but I find that [chat] is the worst way to try and pull someone in because no one's online. And if someone at'd me on Slack, I wouldn't see it if they did at night time. So yeah, basically just sticking to [alerting tooling]"

Recruitment in the other three cases also included examples of recruitment via alerting tooling, but recruitment was primarily done by direct chat messages and chat messages in public channels – these messages are often referred to as a ‘ping’.

> “And so we started ping <team> because they're the ones who are responsible for handling that interface”
> “I knew that if I pinged him, he could get the right engineer involved"
> “I tried pinging <engineer>”

When interviewing a responder about instances of them recruiting others via chat messages, and why this method was appropriate rather than the other mechanisms available to them, the responder stated:

> “if I know that my technique is going to get us a faster response and mitigation than going through [alerting tooling], I'm going to do that”
>
> “If I don't know who I'm talking to or what I'm talking about, I'll find somebody who is familiar and ask 'em to use [alerting tooling] to get somebody in … But if I know who they are, I'll just blast every possible way I can until they pay attention to me.”

There was also one instance of an individual, after failing to reach someone using the alerting tooling, looking up their phone number and calling them directly.

Goals

The research discovered a variety of intentions when recruiting individuals into the response, as stated by individuals during interviews, when asked about instances of recruitment from the chat transcript.

Additional Capacity

People were recruited into the response to help determine the cause of, and fix the issue.

This was at times explicitly stated by the engineer:

CHAT LOG	`“Escalated  to @<engineer> after 35 mins as I couldn't find how to fix”`

INTERVIEW	"I definitely will try and pull in as many people as possible rather than try - who know that service rather than me try and figure it out, which could take all day depending on what it is, it's better to pull someone like a subject matter expert."

As well as not explicitly stated:

CHAT LOG	`"Uhm - this graph is scary @<engineer>"`
INTERVIEW	"Hey <engineer>, this is a really serious issue, can you help investigate?"
	"hey, we've got this serious issue, we need to find what the root cause is. And having multiple people looking reduces the amount of time it takes us to get to mitigation."

Additionally, a route taken for diagnostic help was to go through an intermediary who would help determine the right engineer to contact.

CHAT LOG	`"@<manager> Is there anyone we should bring in to look at <service>"`
INTERVIEW	"I know he's the manager for the <service team>. And so I knew that if I pinged him, he could get the right engineer involved that may have availability to investigate."
	"a manager might know if I don't have a really good personal relationship with the engineer, it's sometimes easier to get the manager to point out who should come look at it"

Engineers also recruited others for help coordinating.

CHAT LOG	`"Created incident facilitator page here for comms <link to alert in paging tooling>"`

INTERVIEW	"the incident facilitator is a person that uses [coordination tool] and then sends out all the emails and stuff and tries to contact people, well not for escalations, but they're supposed to help facilitate the whole incident." "I was kind of busy trying to work out what was going on and also had never done that before, so I just had to get <recruit> in to help."

Cross-Checking

People were also recruited to cross-check decisions.

CHAT LOG	`"I'd be more comfortable with someone from API Platforms weighing in on that."`
INTERVIEW	"I was not completely comfortable just saying definitively, this is the way to do it. So I was hoping we could get somebody on that team involved in the calls so that we didn't make any mistakes"

In this case, a manager had suggested a course of action based on their prior experience managing that team. However, they felt their knowledge had potentially become out of date and were seeking some validation on their suggested approach.

People were also recruited to provide information or provide guidance for decision making.

CHAT LOG	`"@<engineer> Do you know who typically handles executive comms? I'm wondering if we need to send something out for the employer funnel at this point`
	
	`CC <engineer2> <manager>"`

INTERVIEW	"it's not a process I'm actually super familiar with, but I've seen that when we have events that are of such scale, we have to send out some sort of communication" "So I'm asking who usually sends this because they would know if we should."

Authority

People were also recruited for approval on decisions deemed within their scope.

CHAT LOG	`"Do we have anybody from product management here? Especially [part of product]? I'm going to pull someone in. I feel like we need to… make a call as to whether we just want to put up a, sorry, something's happening banner."`
INTERVIEW	"I want to throw up a page, but I need somebody to say yes" "If you don't say no, if you don't give me anything, I'm going to do it anyway. If you say no, I won't, but otherwise it's going to happen. So there was a quick conversation and they were like, okay, fine. Eventually they were just like, just go do what you need to do."

In this example, a responder from engineering, felt another part of the organization (Product) was best equipped to make a decision to help mitigate the impact of the incident. In the interview they explain in more detail:

> "First off, I wish just in general, I wish [Product Managers] would come more often to these firefights. It just helps 'em understand what engineers, what [Software Engineers] go through and places in which their input is valuable. This is one. So if we're completely blocking [functionality], is it better for our reputation to say, sorry, there's a problem, as opposed to just giving them blank screens?"

Criteria

As interview participants discussed their experience of the event, there were frequent mentions of the perceived level of impact and elapsed time.

> **Chat log:** `Escalated to @<engineer> after 35 mins as I couldn't find how to fix`
>
> FROM INTERVIEW
>
> **Researcher:** "What factors contributed to you calling it there and being like, well I need to escalate?"

Engineer: "Well, as I said previously, if I get to 30 minutes, normally I escalate … I think I read somewhere on one of those runbooks that I was working through that if that pipeline lag was high, that [data used by customers] wouldn't update. So this is probably why I decided to do it then, if that makes sense."

Note: It was confirmed by the researcher that this engineer did use the *runbook* (documentation on procedures to follow when investigating issues), which did include a description of impact very similar to what the engineer describes here.

From another case, recruiting out of a feeling of "spinning our wheels":

CHAT LOG	`"@here I'm throwing up a [video call] for this as we need to figure out what steps we can take right now to mitigate this"`
INTERVIEW	"kind of felt like we were spinning our wheels about trying to understand what the root cause was" "basically it seemed like things weren't progressing fast enough"

The @here in the above message is a way of notifying everyone that is in the current channel – in this case, a dedicated channel for the incident. However, there were people in the incident channel who had not been engaging, and so from the perspective of this engineer, had not yet joined the response.

FROM INTERVIEW

Researcher: "So here where you said at here - so you at mentioned the entire channel, the event channel: 'I'm throwing up a [video call] for this as we need to figure out what steps to take to mitigate it.'. Who are you expecting to join that?"

Engineer: "I want anyone that might be lurking in the channel that might have context that's not speaking right now. Because with events people start joining the channel but then they don't say anything. They're just sort of observing and that's sort the context that's missing. We had someone from <team> who joined and wasn't really talking for a good bit, but then it was like, hey, we actually did do a deploy around the same timeframe and that was sort of the thing that the [video] chat can facilitate."

Sources of recruitment information

When describing how they knew to go to a particular individual or team, many of the responses were based on prior experience in the organization.

- "I know a lot about the systems involved in this and who works on 'em."
- "Because I used to work on them, this product"
- "Just working with him. I have been in many meetings, talking about reliability, been in some events, had personal conversations with him. I was on a team with him for a short while."
- "<Manager> and I have a personal relationship. I was his grand boss for a while"

When considering alternatives to who could be recruited, or when attempting to pull in multiple people at the same time, organizational structure was a useful source of information.

- "You've got <manager1> on the [feature 1] side, you've got <manager2> on the [feature2] side, but the next person up is <manager3>. So that was just sort of an org chart awareness"
- "I know he's the manager for the [platforms], API team."
- "maybe <manager> who's his manager, might can make a call about it too. So it's basically rolling up the chain I guess."

Documentation was also used in directing people to those who may be able to help.

- "Yes, because [the documentation] is supposed to be this kind of source of truth for everything. So I was like, now that I know that that service, I'll find it in [the documentation], find who owns it and see if that team has any escalation policies or any way for me to page them."
- "Look up their name in [alerting tool] and then go to [human resources system] to look up a phone number."
- "he is listed as emergency responder [in the alerting tool], so that's why I went to him."

Impediments

Recruitment tooling was observed as an impediment in these cases, either in that information in the tool was unclear/out-of-date, or the tool was not working as anticipated.

- "One problem have I've found is that the [alerting tool] service names - the escalation policies are not necessarily blatantly obvious. Does that make sense? So it's the service, but it's like then it's a step of being like, wait, what policy maps to this service? And trying to figure out that in order to figure out who to page."

- “well, I knew he was not on the incident and was digging through [alerting tool] trying to figure out why we hadn't gotten him on there yet. So I think at that point I'd realized that the pages were not actually going to him.”
- “I think the reason that I was struggling to find the team to page for that was because I was probably looking for [Team X]. I was looking for the [application mentioned in the alert], and I was looking for [Team X], and then eventually I know where that came from because I went into [the documentation] and found the team that owns [the application mentioned in the alert], and it was [Team Y]”
- “That is a rough process. They often page the wrong group.”

And a difficulty was of course, trying to recruit someone, but not getting a response.

- “Yeah, I don't think he answered the [page], I'm pretty sure he didn't even respond to it. No, I notified him a bunch of times.”
- “doesn't necessarily mean that they'll answer because they might be in a different time zone or something”

Discussion

This research explored recruitment in the context of software incident response, presenting a view of "how" and "why" this occurs. This study provides a concrete and grounded description of how this activity unfolds "in the wild", in contrast to the simple frameworks and linear models of incident escalation and reassignment for diagnosis that are commonly referenced in software organizations today (Palilingan & Batmetan, 2018; Chen et al., 2019; Aguilar, 2023; Barash et al., 2007).

Recruitment as Requisite Variety

Observed in these four cases was a diverse display of how individuals manage complexity by bringing new skills and functions to bear.

In attempting to maintain operations, individuals sought help from each other for a variety of reasons: diagnosis, repair, coordination, cross-checking, information gathering and approval. They recruited others to help resolve the issue, but also to mitigate impact or spread awareness that an issue was occurring. They did so with explicit requests for help or by sharing artifacts that they anticipated their colleague would interpret in a way that resulted in them joining the response. They engaged help using a variety of mechanisms, leveraging what they believed would get the most efficient response, which differed from person to person and was influenced by who was being recruited. Recruitment occurred across a range of roles – engineering, product, individual contributors and management, and requests for help occurred throughout the response.

People also employed various strategies in their attempts to recruit others, such as communicating urgency efficiently when recruiting by sharing a "scary" graph, leveraging the organizational structure to identify who might be available to help and using alerting tools in tandem with notifications within their communication platform to help recruits find where they were needed quickly.

Klein (1992) notes of fire fighting, that it may seem "a simple domain, with a well-defined goal: to put fires out" (p. 172), but in reality, the goals are varied:

> In arriving at the scene it may not be clear whether there is smoke (which is dangerous itself) or flames? If there are flames, should the commander prioritize searching for occupants, put the fire out, or prevent it from spreading? (Klein, 1992, p. 172)

It would not be surprising if response to "fires" in software systems are often viewed from a similarly simplified frame, with one well-defined goal. However, observations in this research affirm a more varied and nuanced set of goals from the perspective of responders.

The four cases in this research demonstrate how successfully managing a system with a variety of potential states requires an equally diverse set of response strategies. The variety of goals and methods observed, such as pinging colleagues directly or asking for help in large shared channels when you don't know specifically who to ask, are examples of how people successfully adapted to meet uncertainty and surprise.

When managing complex software systems, unthought-of states present themselves as a regular occurrence. This study is in line with prior research, that a diverse collection of people and teams with a variety of skills, experiences and perspectives that can be brought in to match such variety is an asset – the more complexity, the more diversity required to manage it (Hong & Page, 2004; Cilliers, 2010).

Tacit Knowledge

> "Something in my brain told me" - Responder, Case D

When recruiting others in support of any of these goals, how do individuals know who to recruit? Fundamentally, this is knowledge. Knowledge being defined in a business context as "information that is relevant, actionable, and based at least partially on experience" (Leonard & Sensiper, 1998, p. 113). Knowledge of who to recruit – as observed in this research – exists on a spectrum, from tacit to explicit. In many cases this knowledge was tacit, held entirely in the person's head, resulting from past experience. In other instances, it had been made explicit in the form of documentation or having been codified into tooling.

In one case involving explicit recruitment knowledge, an engineer had received an alert from a tool that provided the ability to escalate to another person in the organization with more domain knowledge of the impacted service. This is explicit in that someone in the organization had configured the tool to provide this escalation point, defining who should be called for help if the need arose. Also within this case, the engineer had looked up (and struggled to find) documentation to determine who else they could recruit to the response. As an organization changes, this explicit knowledge requires constant effort to maintain and quickly becomes outdated. People leave organizations, new people join, teams are reorganized and technical systems undergo constant change.

In more instances than not, who to recruit was based primarily on tacit, rather than explicit knowledge. Working on the same team, having worked together previously, having attended reliability meetings together and having gained exposure to and knowledge of each other's skills and past experience. The individuals interviewed in this research had accumulated these intangible assets over time in their heads and relationships. This was observed both in knowing who can help and going to them directly, as well as not knowing precisely who can help, but knowing who else may have the knowledge of who to call – leveraging tacit knowledge of others.

In all cases, restoring system operations required recruiting additional people into the response, and this recruitment was possible as a result of the tacit knowledge of the recruiters. Even cases supported by explicit recruitment information had tacit dimensions, such as the engineer using prior experience to decide *when* to escalate using the tooling, and knowing how to navigate documentation, weigh the correctness of information, and map information between the documentation and the alerting tool. Explicit knowledge is not sufficient in itself to support the adaptation required to operate complex systems, nor is it possible to make all tacit knowledge explicit, simply by the fact that "we can know more than we can tell" (Polanyi, 1967, p. 4).

Recruitment Expertise as a Critical Resource

> Expertise is a key resource in any organization, but it is usually not treated with the same care as other resources. Few organizations have any methods for preserving or expanding their experience, or even taking stock of their current expertise. (Klein, 1992, p. 170)

Based on observations of recruitment in this research, people's understanding of their colleagues' responsibilities, interests, knowledge and abilities helps the system handle unforeseen surprises. Recognizing that a situation needs assistance, who can help, and how and when to recruit this help is itself expertise. This expertise is as necessary to successfully operate complex software systems as the skills being recruited.

As systems change over time at both a people and technological level, the individuals needed to accomplish the variety of recruitment goals also change. Additionally, people may leave the organization, taking with them an understanding of the system and a history of experiences with technology, people and teams across the organization.

The role of this expertise in helping maintain operations by way of effectively recruiting others may be an under-appreciated or overlooked impact of staff turnover. Unattended, loss or degradation of this expertise may not be noticed until the organization experiences significant and costly impacts.

This risk may be partially mitigated by experts improving or influencing documentation or rules coded into software systems, among other knowledge management approaches (Hislop et al., 2018). However, as mentioned previously, people know more than they can tell, explicit knowledge requires effort to maintain – quickly becoming out of date – and all knowledge has some tacit dimensions, such as behavioral norms around how and when to use it.

Given the significance of recruitment expertise in maintaining operations, it would be beneficial for organizations to foster conditions which enable preservation and expansion of expertise. Approaches exist that may aid in accelerating the development of tacit knowledge, even for individuals who did not have direct experience (Hoffman et al., 2013) – continuous practice of which would offset the impact of losing individuals replete with this knowledge.

> "…the most experienced personnel in an organization are repositories of stories."
> (Klein, 1992, p. 181)

People develop expertise over time. Experts have a deep repository of stories about the organization – who was involved, what people were seeing, key decisions and lessons learned. These stories, when shared and heard or read by others can enable development of expertise vicariously. Stories can be "ideal carriers of tacit dimensions of knowledge" (Swap et al., 2001) because of the rich context provided. Stories are also vivid, engaging and entertaining, thus more memorable and easily retrieved (Swap et al., 2001).

Accounts of non-routine incidents, when collected and shared, can increase the ability of an organization to adapt to challenging events. Stories "function like a voice of experience so that even when the people with expertise are not present, the stories… are still available to remind decision makers that 'This is like the time when _ _ _'" (Klein, 1992, p. 181). Having these stories easily searchable and using them regularly – perhaps as training material for new employees – will help ensure these "lessons learned" don't quickly become "lessons forgotten" (Hoffman et al., 2013, p. 130).

Future Directions

Recruitment as an activity during anomaly response is an understudied topic. This research is an initial exploration into the "how" and "why" of recruitment in operating software systems, and the researcher hopes for this to be encouraging of future research. Dalal, et al. (2016) have similarly observed recruitment, specifically for decision-making purposes, as an understudied decision making structure in cyber-security, and provide an agenda for future basic research in the area.

Questions that could provide future directions include:

- Communication within chat software was heavily used for recruitment. What aspects make this an effective tool for recruitment, and how might existing recruitment tooling be improved, or new tooling be developed to support the variety of goals and methods of recruitment observed in this research?

- Tools and documentation that specify who to recruit for a given application or scenario quickly become out of date, and were observed as an impediment to recruitment in this study. Specifically, what parts of this documentation becomes outdated, and what has changed that makes it no longer relevant? What might be possible methods to keep this information more up to date with less effort?
- During case selection, the research observed that recruitment when responding to software incidents often cascades. Meaning, an individual recruits help, then this recruit pulls in additional responders, who then recruit yet more help, for other reasons. How do the goals and methods of recruitment differ between these layers of recruitment?
- In the cases of this study, many instances of recruitment were unsuccessful. Do people try again, do they use alternative methods, do they look for alternative options for help and does their goal for recruitment change? What criteria influence these decisions?
- What differences in recruitment methods and strategies can be observed between individuals with short and long tenure at an organization?

Conclusion

Allspaw (2015) reached this meta conclusion in the study of heuristics used by engineers in resolving software outages:

> 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. (p. 66)

Similarly, a conclusion that could be drawn from the present research on recruitment 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 cast of people and supporting roles that are available to them for help.**

Put another way, by an engineer from a case in this study –

> "This is [the] knowledge that you learned that adds to your value as an engineer. This is the stuff that you learn. It's like, I know this person is responsible for this and for me to be effective is to flag it to them."

The ability of people to recruit others when faced with unusual demands influences whether a system can be characterized as resilient, or brittle, and at risk for sudden collapse. Recruitment is also inherently collaborative, beyond the capability of any individual engineer or manager. The findings from these cases converge with the central finding of Bergström (2012), that resilient performance during incidents "can be seen to be an emergent property of the interactions, relations and coordinative strategies amongst a multitude of actors, rather than being the result of heroic achievements or 'correct decisions' on the part of single actors" (p. 71).

Successfully managing complexity is clearly more than a technical or process challenge. The tacit knowledge stored in the heads and relationships of people in software organizations directly influences if the system can withstand challenges. This knowledge can also not be made entirely explicit, or easily shared. Preventing catastrophe requires an awareness of what people in an organization know about what other people know, and cultivating the development of this knowledge.

Where technical and process changes can be quickly and easily replicated between organizations, the existence of collective knowledge cannot. Intentionally fostering conditions for this knowledge to develop by investing in practices such as incident storytelling will support efforts to maintain operations, and create a competitive advantage that is not easily replicated.

Ultimately, it is people that keep systems running. People do not do it alone, they develop and leverage expertise that is knowing who and when to ask for help.

Appendix A - Coding Schema

Top Level	Code	Description	Example(s)
Recruiting	Direct	Recruiting a specific person, team, or organization by name	*- I paged the API team to join the incident* *- I asked Eva who could help me here*
	Indirect	Recruiting others without expressed intent for any specific person, team or organization	*- I posted a thread in a shared channel with observations of what I was seeing*
Criteria	Issue not yet diagnosed	An expression that a cause has not yet been identified, or there is not a clear lead or feeling that the incident is progressing towards resolution.	*- Felt like we were spinning our wheels* *- We didn't have any good leads*
	Perception of potential impact	Acknowledgement of the potential user or financial impact of this event	*- This is getting really bad* *- The website is completely down* *- This is affecting revenue*
	Time element	A reference to time when describing their experience or decision making during this event	*- We're not getting to resolution fast enough* *- It has been 30 minutes so I'm escalating*
	Requires skill or knowledge	An expression that some specific thing needs to be done, but the person does not have the skill or knowledge of how to take that action	*- I don't know how to do this* *- I don't know where to look to get that information*
Recruitment goals	Seeking additional capacity	Recruiting for help with tasks (diagnosis, repair, coordination, communication), or to help recruit others	*- I need someone to help look at this and fix it* *- They're supposed to help facilitate the whole incident* *- They can find the person I need to be talking to on their team*
	Cross-checking	Recruiting for help cross-checking decisions, providing information	*- I'm wondering if we need to send out comms* *- I'm not completely comfortable saying this is how we should do it*
	Seeking authority or	Recruiting for approval to take some action	*- I need somebody to make a call*

	approval		
Sources of recruitment information	Using prior knowledge or experience	Deciding who to recruit based on prior knowledge, experience or personal relationships	*- Tribal knowledge* *- I was on a team with them in the past* *- I know a lot about the systems involved and who works on them*
	Documentation	Deciding who to recruit based on explicitly named individuals listed in documentation	*- They were listed as emergency responder for that service, so I went to them* *- I looked in the service directory for who owns that service*
	Using organizational structure	Deciding who to recruit based on position in the organizational hierarchy	*- The next person above them is Jane, so I went to them* *- They are the manager of that team*
	Searching chat history	Deciding who to recruit based on searches through chat history	*- I searched through chat messages to see who has done this before*
Recruitment difficulties	No response	Person being recruited does not respond	*- They didn't answer/respond* *- The pages never got to them*
	Information in tooling unclear or out of date	Difficulties determining who to recruit, when using tooling/process documentation	*- It's not clear how to figure out who to page* *- There was some disconnect between the name of the service and how it appeared in the tooling*

Appendix B - Tools Used in Analysis

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

Jeli

Jeli (https://jeli.io) is an incident management platform with a suite of features that support the analysis of software incidents. Jeli was used in this research for case selection and generating an initial coding scheme. Case data was exported out of Jeli into raw JSON files. These JSON files were then manipulated using Python (https://www.python.org) scripts developed by the researcher for the purposes of this research.

NVivo

NVivo (https://lumivero.com) is qualitative data analysis software used in this research to code transcripts and perform inter-rater reliability tests.

Observable

Observable (https://observablehq.com) is a web-based tool for generating visualizations with code. Observable was used to create most of the figures included in this thesis.

Rev.com transcription service

Rev (https://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.

Appendix C - Informed Consent

Research Project Title:
Adaptive Tactics in Software Operations: How People Manage Complexity by Asking for Help

Student Investigator:
Michael Wettick

Project Purpose and Procedure:
This research will focus on recruitment activities by individuals and teams involved in resolving software issues, aiming to better understand the decision-making and communication involved. Technical data will be gathered (graphs, logs, chat transcripts) as well as interviews done post-event.

Confidentiality:
Identities of all participants will remain anonymous and will be kept confidential from all other parties other than the interviewer. Notes will be taken during the interview for the purpose of recall by the researcher for future analysis. Anonymity will be further protected in any future portions of the thesis paper and any presentations that may result from this work. Participant names will be kept securely and separate from the information collected by the researcher. Participant names and researcher notes will be disposed of using confidential waste no later than March 31, 2024.

Contact Information about this Thesis Work:
If you have any questions whatsoever about the research project, you can contact Michael Wettick, <email>, <phone>.

Risks/Benefits:
There are no known risks or benefits to participating in this research.

Consent:
Your signature indicates that you have received a copy of this consent form for your own records and that you consent to (a) participate in this research; and (b) to allow Student Investigator to use your responses in any way related to the Research Project without providing you with any compensation. To the extent you disclose any Confidential Information as defined in the Non-Disclosure Agreement between Student Investigator and Indeed, that information remains confidential and subject to the terms of the Non-Disclosure Agreement.

I, ____agree to participate as outlined above.

____

Participant's Signature Date

____

Student Investigator Signature Date

References

Aguilar, A. (2023). Lowering Mean Time to Recovery (MTTR) in Responding to System Downtime or Outages: An Application of Lean Six Sigma Methodology. In *13th Annual International Conference on Industrial Engineering and Operations Management.*

Alderson, D. L., & Doyle, J. C. (2010). Contrasting views of complexity and their implications for Network-Centric infrastructures. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, *40*(4), 839–852.

Allspaw, J. 2015. *Trade-offs under pressure: heuristics and observations of teams resolving Internet service outages*. Master's thesis. Lund, Sweden: Lund University.

Allspaw, J. (2022). Critical digital services: An Under-Studied Safety-Critical domain. In *Managing future challenges for safety: Demographic change, digitalisation and complexity in the 2030s* (pp. 33–40). Springer International Publishing Cham.

Anders, S., Woods, D. D., Wears, R. L., Perry, S. J., & Patterson, E. (2006). Limits on adaptation: modeling resilience and brittleness in hospital emergency. *Learning from Diversity: Model-Based Evaluation of Opportunities for Process (Re)-Design and Increasing Company Resilience*, 1.

Ashby, W. R. (1958). Requisite variety and its implications for the control of complex systems. Cybernetica 1 (2): 83-99.

Bainbridge, L. (1979). Verbal reports as evidence of the process operator's knowledge. *International Journal of Man-Machine Studies,* 11(4), 411–436. https://doi.org/10.1016/s0020-7373(79)80035-8

Barash, G., Bartolini, C., & Wu, L. (2007, May). Measuring and improving the performance of an IT support organization in managing service incidents. In *2nd IEEE/IFIP International Workshop on Business-Driven IT Management* (pp. 11-18). IEEE.

Barton, M. A., & Kahn, W. A. (2019). Group resilience: The place and meaning of relational pauses. *Organization Studies*, 40(9), 1409-1429.

Bergström, J. (2012). *Explorative studies of high-risk situations from the theoretical perspectives of complexity and joint cognitive systems* (Doctoral dissertation, Doctoral thesis. Media-Tryck, Lund University, Lund, Sweden. Report).

Beyer, B., Jones, C., Petoff, J., & Murphy, N. R. (2016). *Site reliability engineering: How google runs production systems.* O'Reilly Media, Inc.

Blaxter, L., Hughes, C., & Tight, M. (2010). *How to research*. Berkshire, England: McGraw-Hill – Open University Press.

Chen, J., He, X., Lin, Q., Xu, Y., Zhang, H., Hao, D., Gao, F., Xu, Z., Dang, Y., & Zhang, D. (2019). An empirical investigation of incident triage for online service systems. *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, 111–120.

Cilliers, P. (2010). Difference, identity and complexity. In *Complexity, difference and identity: An ethical perspective* (pp. 3-18). Dordrecht: Springer Netherlands.

Cook, R. I. (2006). Being bumpable. In D. D. Woods & E. Hollnagel, *Joint cognitive systems: Patterns in cognitive systems engineering*. CRC Press.

Cook, R. I. (2020). Above the line, below the line: The resilience of internet-facing systems relies on what is below the line of representation. *Queueing Syst.*, *17*(6), 41–51.

Cook, R. I., & Long, B. A. (2021). Building and revising adaptive capacity sharing for technical incident response: A case of resilience engineering. *Applied ergonomics,* 90, 103240.

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

Creswell, J. W., & Creswell, D. J. (2017). *Research design: Qualitative, quantitative, and mixed methods approaches*. SAGE Publications.

Crotty, M. J. (1998). The foundations of social research: Meaning and perspective in the research process. *The foundations of social research,* 1–256.

Cuvelier, L., & Falzon, P. (2017). Coping with uncertainty. resilient decisions in anaesthesia. In *Resilience engineering in practice* (pp. 29–43). CRC Press.

Dalal, R. S., Bolunmez, B., Tomassetti, A. J., & Sheng, Z. (2016). Escalation : An Understudied Team Decision-Making Structure. In *Psychosocial Dynamics of Cyber Security* (1st Edition, pp. 74–91). Routledge.

Ericsson, K. A., & Simon, H. A. (1980). Verbal reports as data. *Psychological review*, 87(3), 215.

Grayson, M. R. (2018). *Approaching overload: Diagnosis and response to anomalies in complex and automated production software systems* (Master's thesis). The Ohio State University.

Griffin, M. A., Neal, A., & Parker, S. K. (2007). A new model of work role performance: Positive behavior in uncertain and interdependent contexts. *Academy of management journal,* 50(2), 327-347.

Hern, A. (2021, June 9). Fastly says single customer triggered bug behind mass internet outage. *The Guardian*. https://www.theguardian.com/technology/2021/jun/09/fastly-says-single-customer-triggered-bug-that-caused-mass-outage

Hislop, D., Bosua, R., & Helms, R. (2018). *Knowledge Management in Organizations: A critical introduction* (4th ed.). Oxford University Press.

Hoffman, R. R., Feltovich, P. J., Dibello, L., Fiore, S. M., & Andrews, D. H. (2013). *Accelerated expertise: Training for high proficiency in a complex world*. Psychology Press. https://doi.org/10.4324/9780203797327

Hofmann, D. A., Lei, Z., & Grant, A. M. (2009). Seeking help in the shadow of doubt: The sensemaking processes underlying how nurses decide whom to ask for advice. *Journal of Applied Psychology*, 94(5), 1261.

Hollnagel, E. (2012). Coping with complexity: past, present and future. *Cognition, Technology & Work*, 14, 199-205.

Hong, L., & Page, S. E. (2004). Groups of diverse problem solvers can outperform groups of high-ability problem solvers. *Proceedings of the National Academy of Sciences of the United States of America*, *101*(46), 16385–16389.

Jansen, I., Stalmeijer, R. E., Silkens, M. E., & Lombarts, K. M. (2021). An act of performance: Exploring residents' decision-making processes to seek help. *Medical education*, 55(6), 758-767.

Jeli [Computer Software]. (2024). Retrieved from https://jeli.io

Jilani, H. (2021, October 7). For millions, the WhatsApp outage could have been a matter of life and death. *The Guardian*. https://www.theguardian.com/commentisfree/2021/oct/07/whatsapp-outage-matter-life-death-social-media-facebook

Kahn, W. A., Barton, M. A., Fisher, C. M., Heaphy, E. D., Reid, E. M., & Rouse, E. D. (2018). The geography of strain: Organizational resilience as a function of intergroup relations. *Academy of management review,* 43(3), 509-529.

Klein, G. (1992). Using knowledge engineering to preserve corporate memory. In *The Psychology of Expertise* (pp. 170–187). Springer New York.

Klein, G. (1993). A recognition-primed decision (RPD) model of rapid decision making. In *Decision making in action: Models and methods*, *5*(4), 138–147.

Klein, G. (2008). Naturalistic decision making. *Human factors*, 50(3), 456-460.

Klein, G., Pliske, R., Crandall, B., & Woods, D. D. (2005). Problem detection. *Cognition, Technology & Work*, 7, 14-28.

Lanir, Z. (1986). Fundamental surprise. *Decision Research*, 424.

Lee, F. (1997). When the going gets tough, do the tough ask for help? Help seeking and power motivation in organizations. *Organizational behavior and human decision processes*, 72(3), 336-363.

Leonard, D., & Sensiper, S. (1998). The role of tacit knowledge in group innovation. *California Management Review*, *40*(3), 112–132.

Leveson, N. G. (1986). Software safety: Why, what, and how. *ACM Computing Surveys (CSUR)*, 18(2), 125-163.

Lund University. (2023, December 14). Research ethics and animal testing ethics. Retrieved May 20, from https://www.staff.lu.se/research-and-education/research-support/research-ethics-and-animal-testing-ethics

Maguire, L. M. D. (2020). *Controlling the costs of coordination in Large-Scale distributed software systems* (Doctoral dissertation).

Maitlis, S. (2005). The social processes of organizational sensemaking. *Academy of management journal*, 48(1), 21-49.

Marsh, Amy V. (n.d.). Accept Help. https://www.yayvm.com/ [Accessed: 2024-01-14].

Nisbett, R. E., & Wilson, T. D. (1977). Telling more than we can know: Verbal reports on mental processes. *Psychological review*, 84(3), 231.

NVivo [Computer Software]. (2024). Retrieved from https://lumivero.com/products/nvivo/

Ormerod, P. (2007). Cascades of failure and extinction in dynamically evolving complex systems. *Noise and Stochastics in Complex Systems and Finance*, *6601*, 160–163.

Palilingan, V. R., & Batmetan, J. R. (2018). Incident Management in Academic Information System using ITIL Framework. *IOP Conference Series: Materials Science and Engineering*, 306(1), 012110.

Polanyi, M. (1967). *The Tacit Dimension*. Doubleday.

Patterson, E. S., & Woods, D. D. (2001). Shift changes, updates, and the on-call architecture in space shuttle mission control. *Computer Supported Cooperative Work (CSCW)*, 10, 317-346.

Perrow, C. (1999). *Normal accidents: Living with high risk technologies*. Princeton University Press.

Sagan, S. D. (1993). *The limits of safety: Organizations, accidents, and nuclear weapons*. Princeton University Press.

Seale, C., Silverman, D., Gubrium, J. F., & Gobo, G. (2003). Qualitative research practice. *Qualitative Research Practice*, 1–640.

Senge, P. M. (2006). *The fifth discipline: The art and practice of the learning organization*. Doubleday/Currency.

Shattuck, L. G., & Woods, D. D. (1997, October). Communication of intent in distributed supervisory control systems. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (Vol. 41, No. 1, pp. 259-263). SAGE Publications.

Swap, W., Leonard, D., Shields, M., & Abrams, L. (2001). Using mentoring and storytelling to transfer knowledge in the workplace. *Journal of management information systems,* 18(1), 95-114.

Van der Rijt, J., Van den Bossche, P., van de Wiel, M. W., De Maeyer, S., Gijselaers, W. H., & Segers, M. S. (2013). Asking for help: A relational perspective on help seeking in the workplace. *Vocations and learning*, 6, 259-279.

Weick, K. E. (1990). Technology as equivoque: Sensemaking in new technologies. *Technology and organizations*, 1–44.

Weick, K. E., Sutcliffe, K. M., & Obstfeld, D. (2005). Organizing and the process of sensemaking. *Organization Science*, *16* (4), 409–421.

Woods, D. D. (1993). Process tracing methods for the study of cognition outside of the experimental psychology laboratory. In *Decision making in action* (pp. 228–251).

Woods, D. D. (1994). Cognitive demands and activities in dynamic fault management: abductive reasoning and disturbance management. *Human factors in alarm design,* 63-92.

Woods, D. D., & Branlat, M. (2017). Basic patterns in how adaptive systems fail. In *Resilience engineering in practice* (pp. 127-143). CRC Press.

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

Woods, D. D., & Patterson, E. S. (2001). How unexpected events produce an escalation of cognitive and coordinative demands. In P.A. Hancock, & P.A. Desmond (Eds.)*, Stress, workload, and fatigue.* Mahwah, NJ: L. Erlbaum.

Woods, D. D., & Sarter, N. (2000). Learning from automation surprises and going sour accidents. In N. Sarter & R. Amalberti (Eds.), *Cognitive engineering in the aviation domain* (pp. 327–354). CRC Press.

Woods, D. D., & Wreathall, J. (2016). Stress-strain plots as a basis for assessing system resilience. In *Resilience engineering perspectives, volume 1* (pp. 157–172). CRC Press.

Yin, R. K. (2018). Case study research and applications: Design and methods (6th ed.). SAGE Publications.