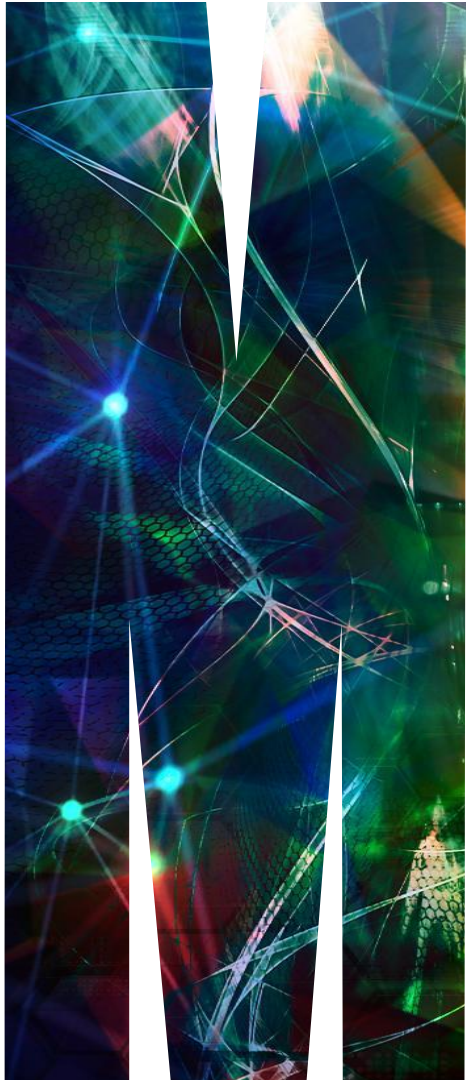
MONASH University

Impact of End User Human Aspects on Software Engineering

ARC Laureate Professor John Grundy

https://www.monash.edu/it/humanise-lab

HUMANISE



Outline

Why END USER human aspects critical to consider during Software Engineering

Examples of end user human aspects and what happens when DON'T adequately consider

Examples of our recent work to improve the situation…

Research Roadmap needed

Summary

Why END USER human aspects critical to consider during Software Engineering

Examples of end user human aspects and what happens when DON'T adequately consider

Examples of our recent work to improve the situation…

Research Roadmap needed

Summary

Human aspects & Software Engineering...

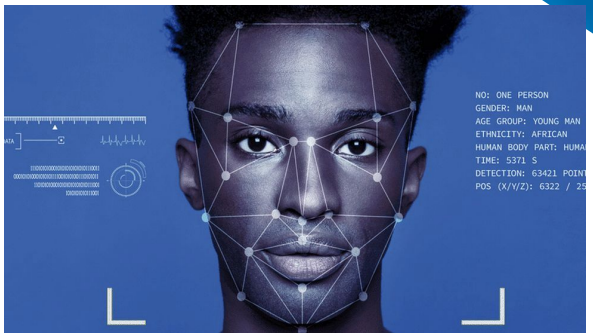
Gender bias – UIs, seat belts, health apps

Ethnic bias – over-recommend minorities for search, don’t recognize faces

Culture bias – inappropriate words, phrases, colours, icons, workflow

Language bias – over-technical, wrong dialect, impersonal

Age bias – too complex, too simple, inappropriate words, symbols, workflow



Human aspects & Software Engineering...

All Can Apply to TEAM and USERS!!!

Physical challenge bias – guesture, sound, sight, voice inappropriate

Cognitive challenge bias – raise anxiety, poor fit to mental model

Enjoyment bias – boring, unengaging, distracting

Emotional bias – stressful, anxiety-inducing, frightening

Personality bias – workflow, lack of engagement, disconnected

And… many others :-(

Why END USER human aspects critical to consider during Software Engineering

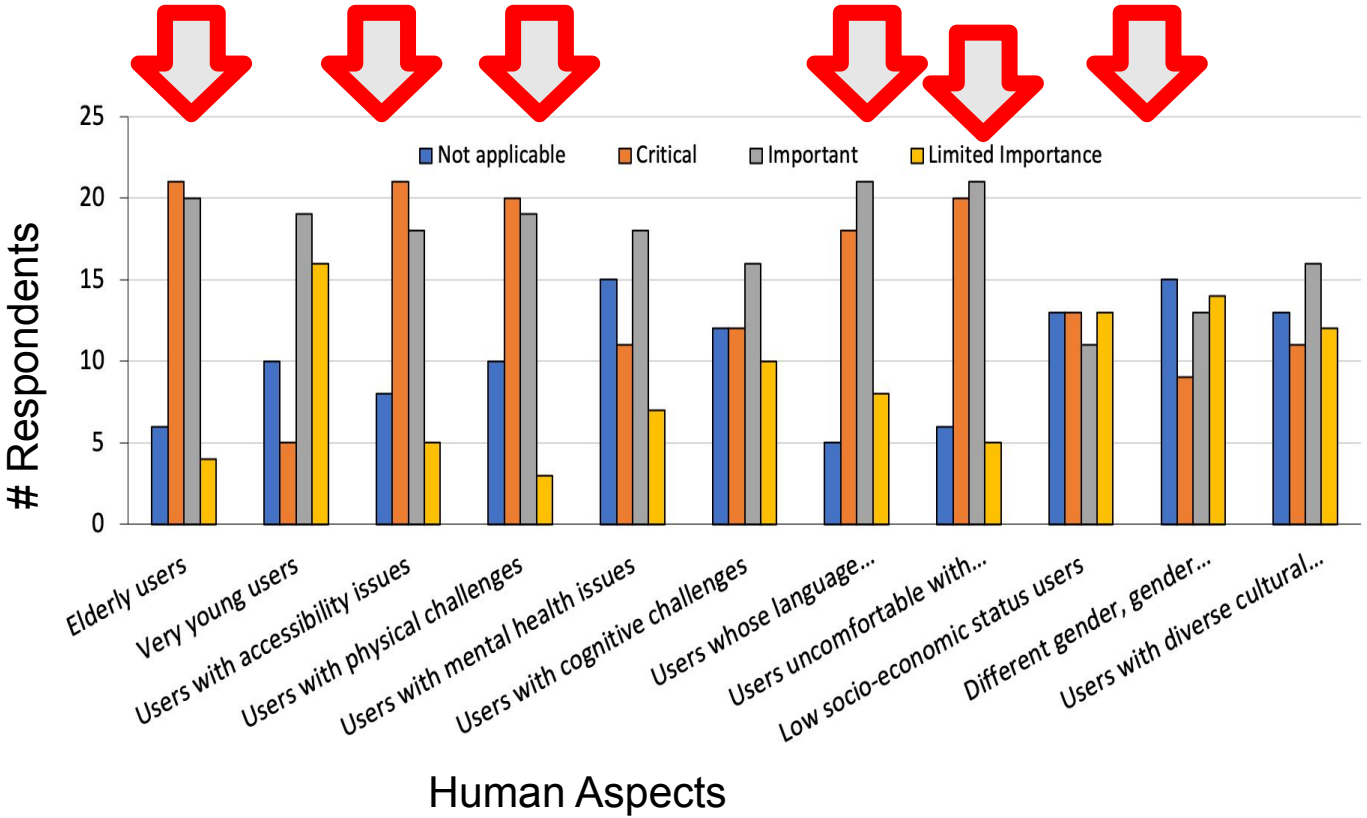
Examples of end user human aspects and what happens when DON'T adequately consider

Examples of our recent work to improve the situation…

Research Roadmap needed

Summary

Findings from recent Developer Survey...



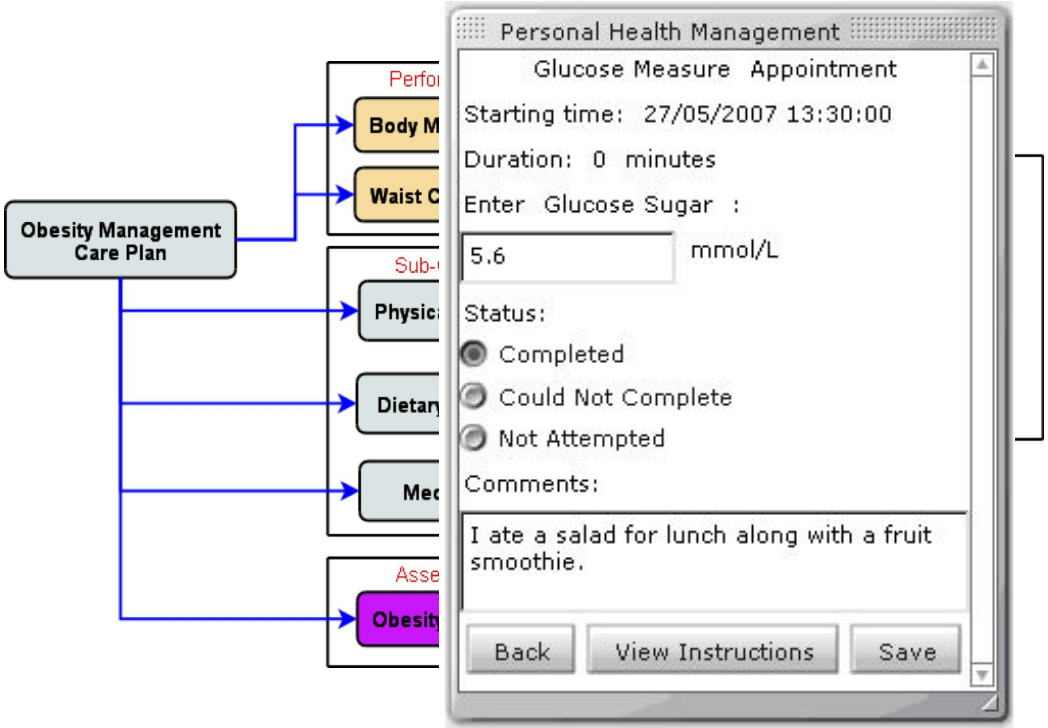
2020 survey we conducted (paper coming! :-))

59 developers, 12 interviewees

Many “critical” and “important” human aspects identified

MONASH University

Need for human aspects - counter example....



Model-driven, end user approach

Clinician models care plan, specialises for patient, generates app for patient

BUT

Fails to take account of ageing patient, gender, culture, language proficiency, terminology, accessibility issues, ...

Why END USER human aspects critical to consider during Software Engineering

Examples of end user human aspects and what happens when DON'T adequately consider

Examples of our recent work to improve the situation…

Research Roadmap needed

Summary

Requirements Challenges

Some of the problems:

- Who are the stakeholders/users we need to take into account?
- How do we elicit/fully consider the human aspects of these stakeholders/users?
- How do we model and reason about their human aspects?

Solutions / research needs:

- Better ways to identify stakeholders, elicit requirements
- New ways to find, extract human aspect-related requirements
  Extend/new domain-specific (visual) languages to model these

Using personas to improve Requirements Engineering

- Software engineers do not understand many critical human aspects
- Example: “Smart” parking app

Need: To quickly find a parking spot at peak hours
Human aspect: Suffers from (red) colour blindness
Issue: App uses the red colour to identify available parking spots

Persona example

Name: Elizabeth Craw

Age: 68

Occupation: Retired

Family: Married, 2 kids, 1 granddaughter

Location: Clayton

Demographics

Goals:

- Wants to visit her children and grandkid every weekend
- Being able to find a parking spot easily even during peak hours
- Be able to bring her husband to the hospital every week
- Be able to use her phone and parking applications despite her vision issues
- Be able to reserve and pay for a parking spot on her phone before reaching her destination.

Key goals

Elizabeth recently retired from working as a counter attendant at Coles in Caulfield. Her two sons live in Melbourne city and she loves to go and visit them every weekend to spend some time with them and her grandchild. She loves travelling to other countries but has been unable to in in the past two years as her husband has fallen sick and she has been taking care of him.

Elizabeth suffers from protanopia (colour-blindness red weakness) and now from a bit of vision impairment but that has not discouraged her from learning to drive since she was young. She loves to be able to move around the city and thus being able to drive was very important for her as she also needs to bring her husband to doctor visits every now and then. However, it has always been a struggle for her to find a parking when she goes to the city especially during busy hours.

Key human aspect considerations

With the rise in technology use in the past decade, her sons have gifted her a smartphone on her 65th birthday. She is a quick learner and has found out how useful a smartphone can be for her. She has tried multiple applications to help her drive around and find a parking spot when needed but none of them had all the functionalities and the support for vision impaired / colour-blind people as she wanted. She once even got fined when using one of those parking applications even though she did nothing wrong, as she misinterpreted a '0' for an 'O' in the parking application when registering her vehicle's plate number for parking.

Key frustrations

MONASH University

Using personas in RE

- Personas give us a way to model and reason about (i.e., “stand in the shoes” of) end users
- Can be used throughout development
- We are interested in:
  - Using personas more widely in requirements definition
  - Supporting persona specification by:
    - providing guidelines
    - persona building tools

Other RE improvements

Exploring new ways to identify “stakeholders” in software projects

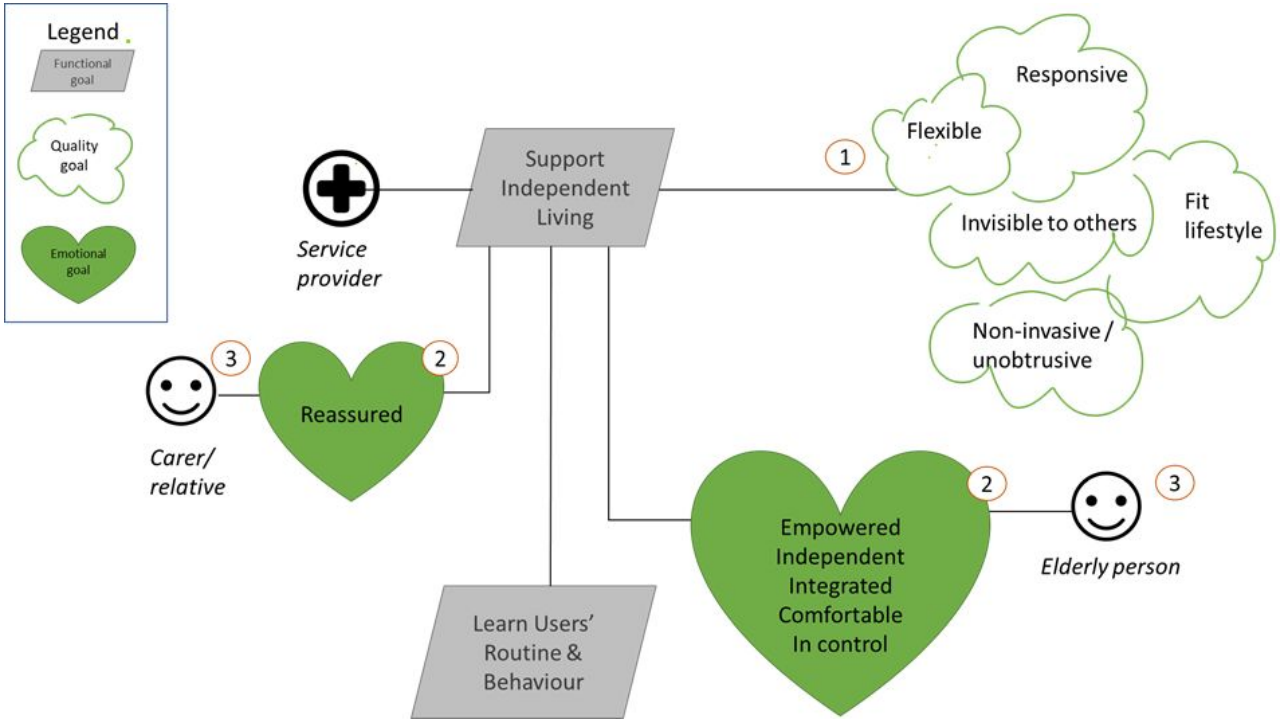
Stakeholders not always users...

Improving dialogue between software engineers and stakeholders to elicit requirements

Capturing human aspects in requirements models

Reasoning about missing requirements, missing human aspects, improperly elicited requirements...

Modelling stakeholder emotions



Design Challenges

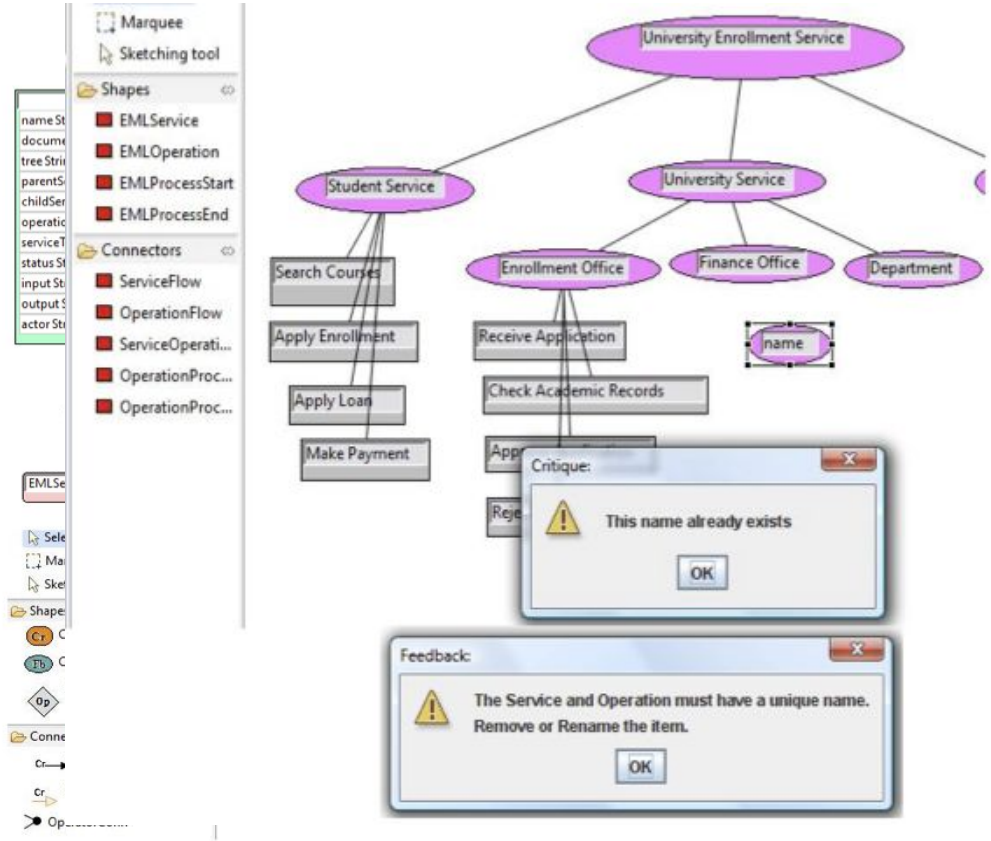
Problems:

- How do we translate human aspects of requirements to designs?
- How do we know these models are complete, correct, effective?
- How do we improve designs to address wide range of human aspects?
- How do we support developers to do this more effectively?

Solutions / research needed:

- Extend design models with human aspects
- “Design critic”-style analysis of requirements and design models
- Better leverage augmented design models, personas during SE
- Provide developers with guidelines, processes, tools to better address end user human aspects in design (and implementation)

Validating human aspects in SE models



- “Design critics” are proactive agents advising designers during design process
- Could advise on missing human aspects, not fulfilled human aspect requirements in design, mis-use of design approaches, trade-offs in design approaches
- Example of critic generator tool at left for adding design critics to DSVL-based tools

MONASH University

Improved designs with variety of personas, extended DSVL models

Different end user human aspects require different design solutions

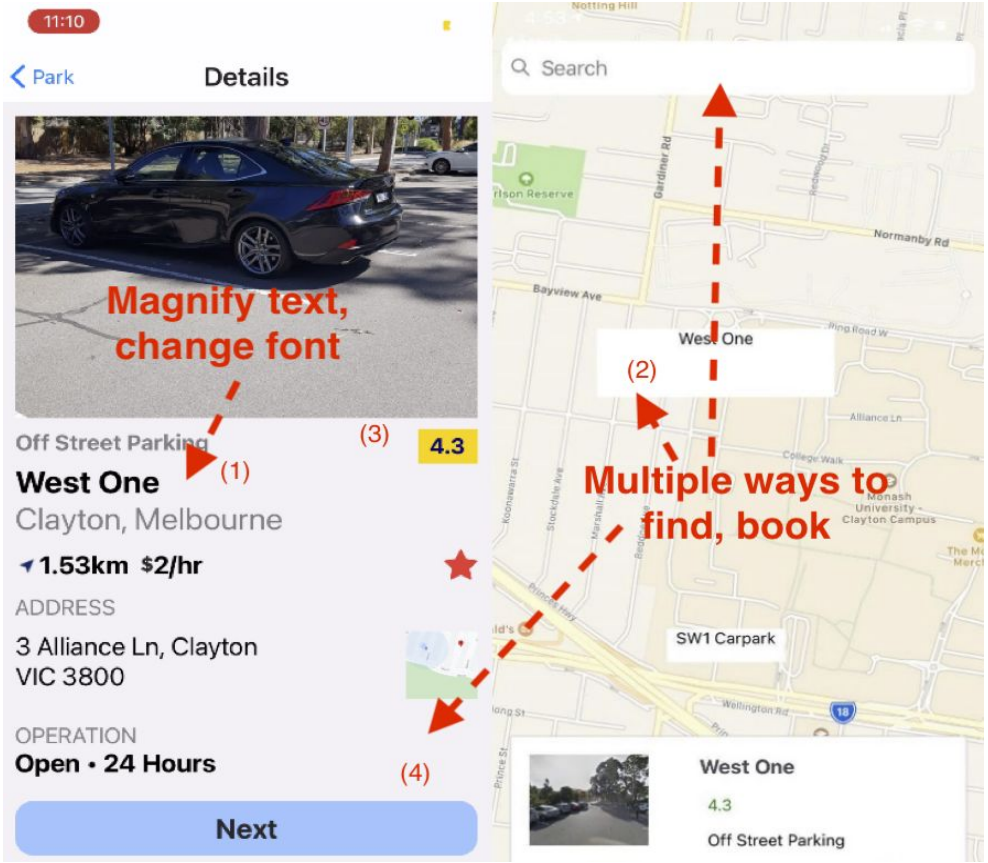
E.g. accessibility challenges => different font, colour, interaction style, voice control, etc needed

Gender => different problem solving styles used means multiple ways to use needed

Culture, language => different metaphors, workflow, terminology, icons, etc needed

Personality, cognitive style => different terminology, dialogue, workflow needed

Better parking app...



Example of “smart parking app” prototype @ left with range of personas, end user differences trying to address vs existing ones

MONASH University

Implementation Challenges

Problems:

- How do we realise different designs for end user human aspects?
- Do we have multiple versions of app vs highly adaptable app or both?
- Can end users change their own apps to better suit them?

Solutions / research directions:

- End user development tools to support end users to build, reconfigure software
- Adaptive user interfaces and associated architectures
- End users specify their preferences for software to incorporate

End user specification, generation of software

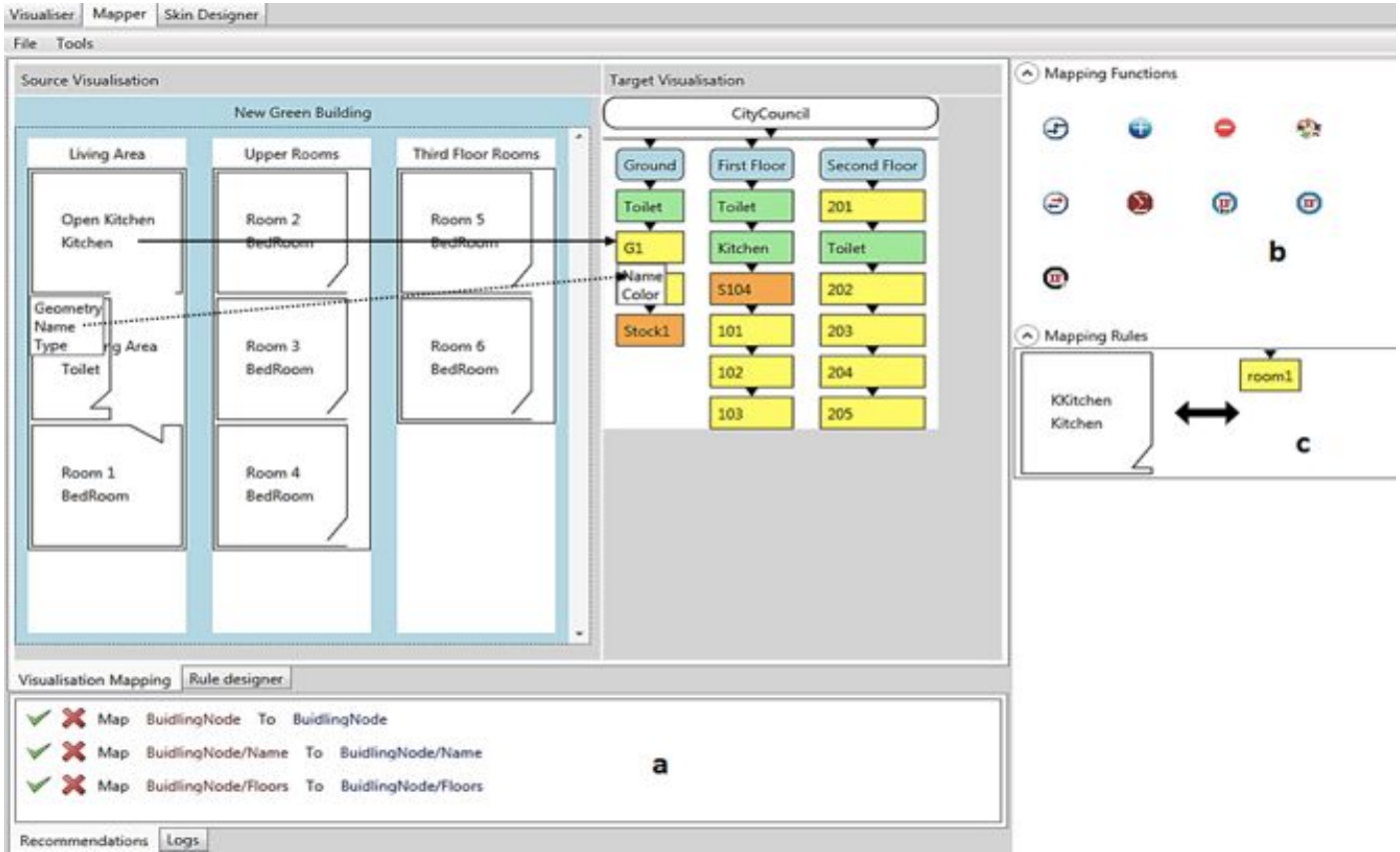
“End User Development” tried over many years to remove software engineers from the process

No code / low code solutions latest attempt...

Often very limited domains / too limited

But allow end users to address their own human aspects proactively

Example: CoNVErT



Example: CoNVErT tool at left for specifying complex data visualisation and data translation software

MONASH University

Adaptive User Interfaces

Adaptive and adaptable user interfaces tried for many years

Often focus on platform adaptation vs end user human aspect adaptation

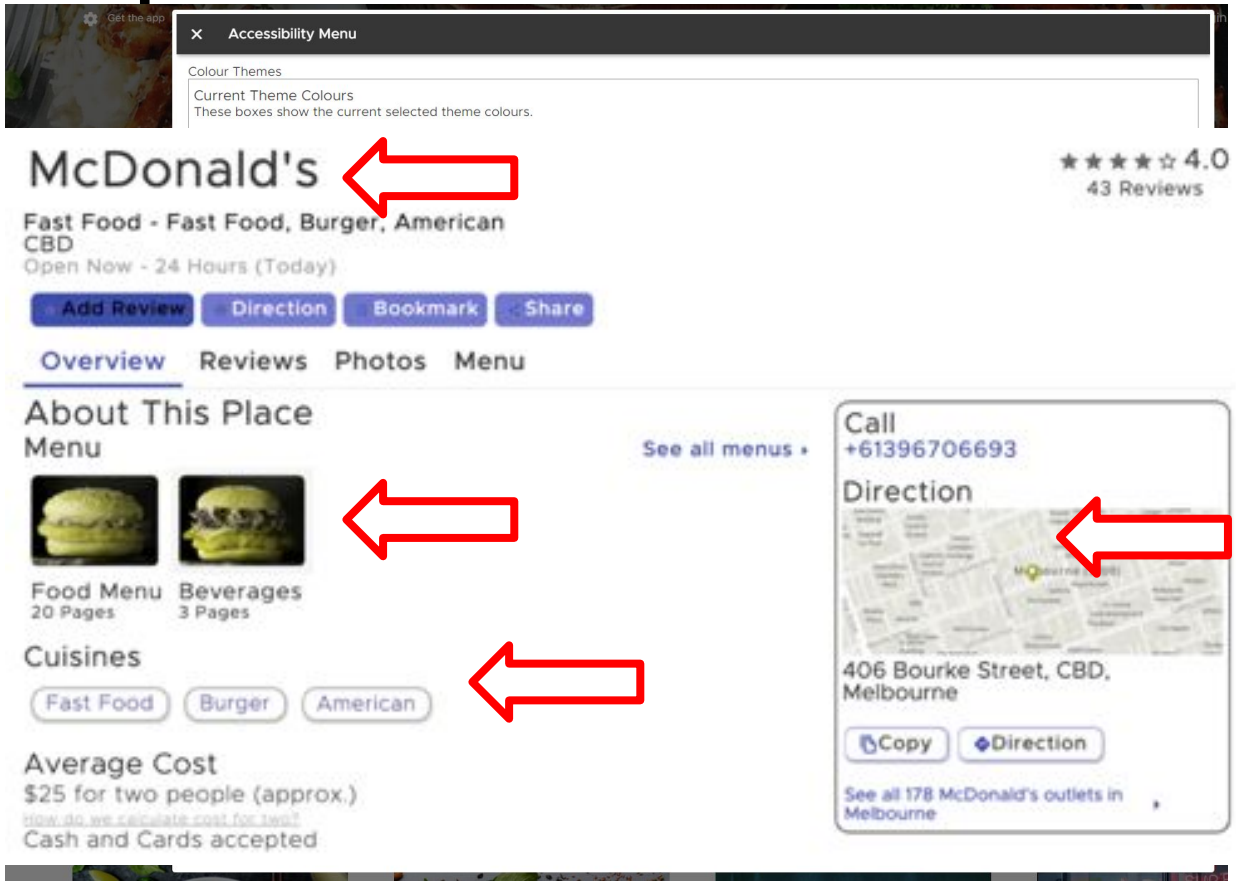
Limited effectiveness

Some AI-based adaptation tried

Want to

- Support multiple different user human aspects e.g. colour blind, no hearing, dyslexic, low motor skills
- Want to allow user to reconfigure how interact with software
- Want software to adapt to end user needs as they become apparent

Adaptive User Interfaces



Example on left of configuring web site for colour blindness, sight challenges, dyslexia, etc

Parking app has similar end user configuration & adaptive UI

MONASH University

Evaluation Challenges

Problems:

- How do end users report human aspect-related defects in software?
- How do we present these human-centric defects to developers to help them understand, appreciate, and suitably fix the defects
- Can we leverage large datasets of user reviews to diagnose and fix human aspect defects in apps?

Solutions / research directions:

- Develop more human-centric defect reporting - better capture defects AND better support diverse end users reporting them
- Human values-based evaluation of app reviews to identify major problems

Human-centric Defect Reporting

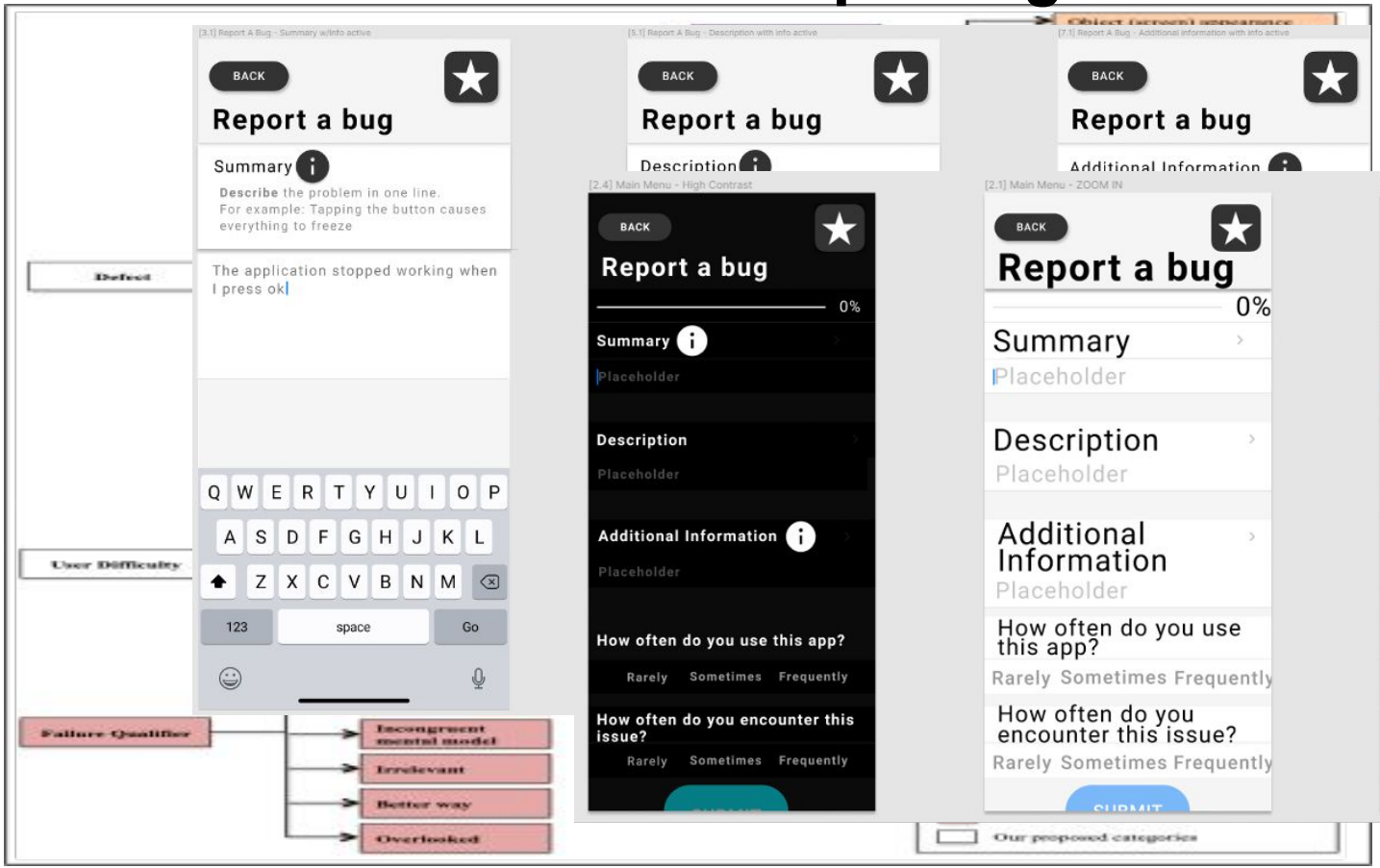
Need improved taxonomy of “human-centric defects” (like our usability defect taxonomy)

Need to use this to guide user to capture sufficient human aspect defect details

Need to make defect reporting tools more accessible to diverse end users

Need to help developers understand better the defects, defect reporter point of view - using personas to represent defect reporters to developers

Human-centric Defect Reporting



Human Values-based app analysis

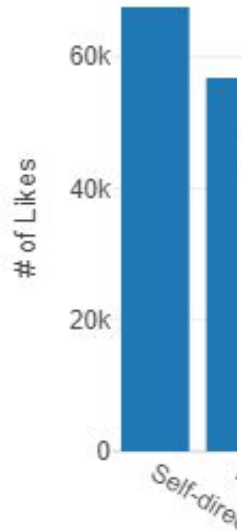
Large app review datasets provide source for rich defect information

Been doing ehealth, COVID-19, social media etc app review analysis

Including variety of human aspects and human values

Example: eHealth app analysis for “human value violations” i.e. violating end user human values such as transparency, privacy, pleasure, capability, ...

Human Values-based app analysis



SAMPLE APP FEATURES AND RELATED VALUE-VIOLATIONS.

Feature	App Name	Value-violation
Save recipes	Pinterest	Curiosity, helpfulness, honesty
Add workouts	TrainingPeaks	Curiosity, helpfulness
Resource pack	Minecraft	Curiosity
Play online	Monopoly	Helpfulness, pleasure
Add stickers	PicsArt	Pleasure
Set reminders	Any.do	Responsibility
Create group	Telegram	Helpfulness, Honesty
Add review	Tripadvisor	Helpfulness, Honesty
Parking history	PayByPhone	Curiosity, Helpfulness
Parking payment	Cellopark	Curiosity
Watch videos	Tiktok	Enjoying life
Tap & pay	CBA	Helpfulness

Why END USER human aspects critical to consider during Software Engineering

Examples of end user human aspects and what happens when DON'T adequately consider

Examples of our recent work to improve the situation…

Research Roadmap needed

Summary

Process challenges

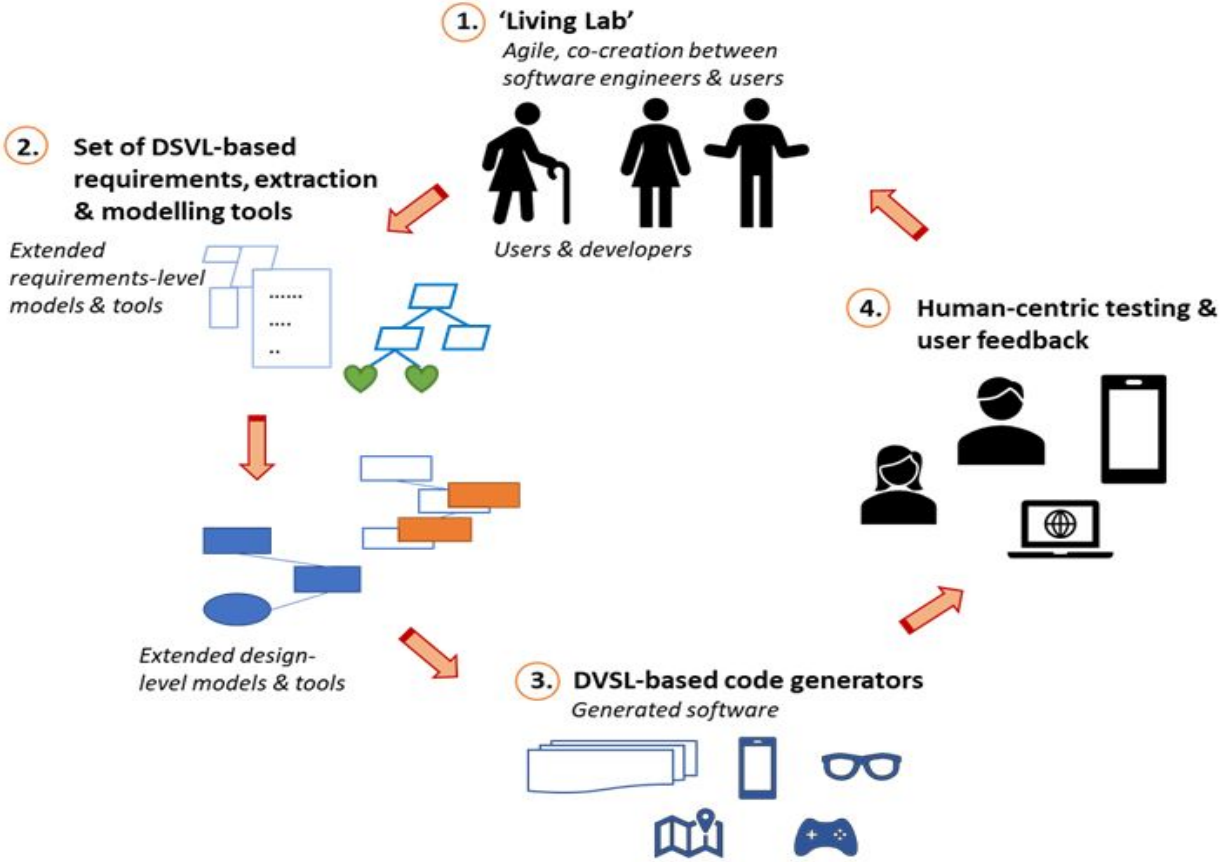
Problems:

- How do we work closely with end users and stakeholders throughout software development to better identify, appreciate and include their diverse human aspects?
- How do we proactively address issues raised by end users?

Solutions:

- Trying out a co-creational “living lab” approach
- Need to prioritise identifying end user human aspects
- Need to train software engineers to see the importance of, better understand, and incorporate end user human aspects in their software solutions

Our approach...



MONASH University

Key things we need to work on

- lack of a taxonomy of end user human aspects including keywords, phrases and examples
- lack of studies focusing on how ***software engineers*** and software engineering teams influence and address end user human aspects in software
- lack of tools to identify challenging end user human aspects to address during requirements engineering, including extraction, modelling, 3Cs checking, and validation
- a range of design and evaluation guidelines and tools but lack of connectivity, consistency, and applicability of these tools in many domains e.g. for mobile app development

Key things we need to work on

- overly-complex, inaccessible and incomplete design and implementation guidelines to address many challenging end user human aspects
- difficulty in end users reporting human aspect defects in software, difficulty in software engineers understanding these defects
- development processes that still don't sufficiently include diverse stakeholder perspectives
- deficiencies in the education of software engineers regarding human aspects of their end users

Summary

Stakeholders and end users of software are very diverse

We currently don’t have good ways to incorporate their diversity into software engineering

Need new approach - avoid “them” vs “us” we currently have

Need ways to fully engage, include end users/stakeholders

Need ways to better capture, model, reason about, design and implement for, adapt, evaluate, receive feedback on and improve software

Software engineers themselves are humans with many diverse human aspects that impact DOING software engineering and WORKING with stakeholders (and each other)...

References

Grundy J., Khalajzadeh H., McIntosh J., Kanij T., Mueller I. (2021) HumaniSE: Approaches to Achieve More Human-Centric Software Engineering. In: Ali R., Kaindl H., Maciaszek L.A. (eds) Evaluation of Novel Approaches to Software Engineering. ENASE 2020. Communications in Computer and Information Science, vol 1375. Springer

Obie, H., Hussein, W., Xia, X., Grundy, J.C., Li, L., Turhan, B., Whittle, J. and Shahin, M., A First Look at Human Values-Violation in App Reviews, 2021 IEEE/ACM International Conference on Software Engineering, online 23-29 May 2021, IEEE

Jim, A.Y., Shim, H., Wang, J., Wijaya, L.R., Xu, R. Khalajzadeh, H., Grundy, J.C., Kanij, T., Improving the Modelling of Human-Centric Aspects of Software Systems, 16th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE2021), online, 26-27 April, 2021

Shamsujjoha, M., Grundy, J.C., Li, L., Khalajzadeh, H., Lu, Q. Human-Centric Issues in eHealth App Development and Usage: A Preliminary Assessment, 28th IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER '21), ERA Track, Online, 9-12 March, 2021

Yusop, N.S.M., Grundy, J.C., Vasa, R., Schneider, J-G, A Revised Open Source Usability Defect Classification Taxonomy, Information and Software Technology, vol. 128, Dec 2020, Elsevier.

Avazpour, I., Grundy, J.C., Zhu, L., Engineering Complex Data Integration and Harmonization Systems, Journal of Industrial Information Integration, vol 16, Elsevier, Dec 2019

Curumsing, M.K., Fernando, N., Abdelrazek, M., Vasa, R., Mouzakis, K., Grundy, J.C. Emotion-oriented Requirements Engineering: A Case Study in Developing A Smart Home System for the Elderly, Journal of Systems and Software, vol 147, Jan 2019, Elsevier, pp. 215-229.

Barnett, S., Avazpour, I., Vasa, R., Grundy, J.C. Supporting Multi-View Development for Mobile Applications, Journal of Computer Languages, Volume 51, April 2019, Elsevier, Pages 88-96

Ali, N.M., Hosking, J.G., Grundy, J.C., A Taxonomy and Mapping of Computer-based Critiquing Tools, IEEE Transactions on Software Engineering, vol. 39, no. 11, November 2013, pp. 1494-1520