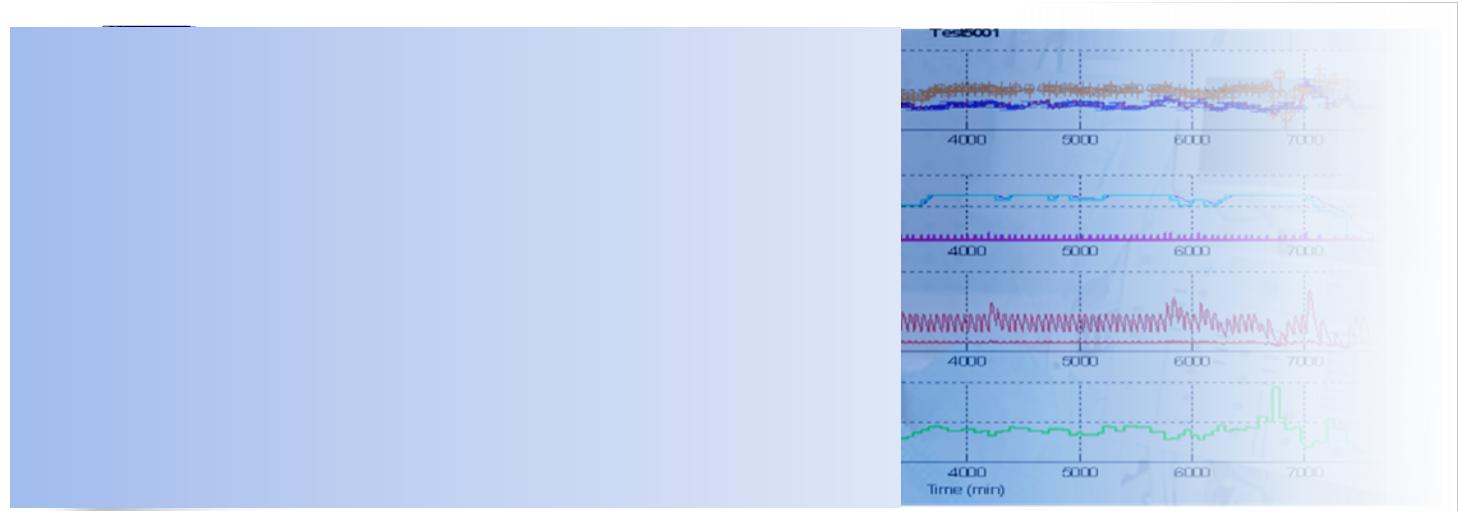


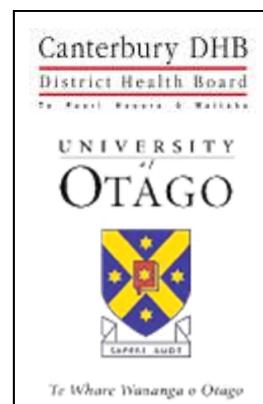
What went wrong? The importance of better human factors



JG Chase, AJ Le Compte, J Steel, LM Ward, A Evans, C Tan, A Lynn and GM Shaw



**Centre for Bio-Engineering
Department of Mechanical Engineering
University of Canterbury
Christchurch, New Zealand**



- 
- Human factors has a science to it but is actually a very experiential field comprising engineers, designers, social scientists and others
 - Some say that the easiest way to learn from the experience of others is via stories, a tradition 1000s of years old
 - So, today we are going to try it, delivering our experiences with human factors via seven (7) hopefully interesting stories from our experience that will hopefully also have relevant aspects to your work
 - Then, like any good fable, we'll attempt to tie it all together in some “morals” and lessons learned.
 - Oh, and just before summing up, I will tell you what human factors is with some proper (and improper) definitions ...!

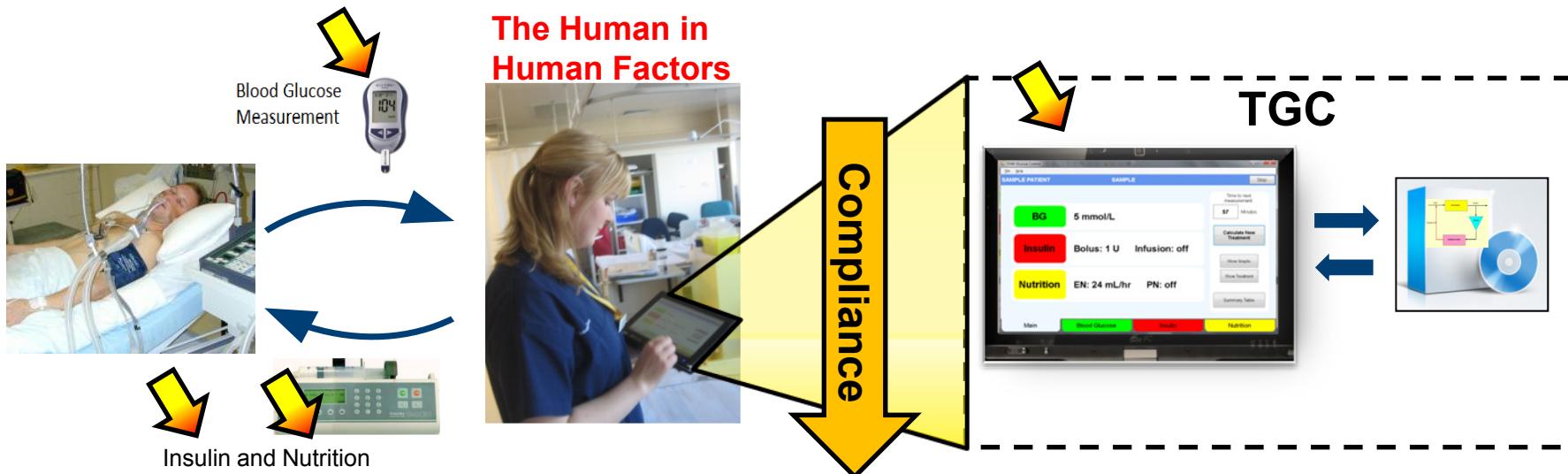
A view of technology use...

Greater use of devices and interfaces to guide therapy selection, both in-hospital and for out-patients

Hospital TGC = 1 example, but the story morals generalise

- Notice all the areas where a human interface is involved!

- Hint: It's all 3 devices and every step within their use
- Without the human there are no human factors



In the beginning ...

- All stories start this way ... and in diabetes technology (in-hospital or out) it goes something like this...
- I/we had a whole bunch of the latest gear (sensors, pumps, a protocol (perhaps computerised)) and a dedicated nursing staff
- We'd spent a lot of time on this and the big day to pilot trial it all finally came ...

(some period of time elapses)

- The results were ok, but not fully satisfactory. Things didn't always seem to go entirely as planned ... Why?
 - In fact, a lot of time in journal papers gets spent on this "why", often related to physiology and other more tangible factors we "know about"

1. The NICU story

- Glycemic control is difficult here and we have a model-based system in regular use in Christchurch after much piloting
 - Give it a BG value and it will recommend an exact insulin dose based on a model-identified insulin sensitivity (most such systems share this essential approach)
- If you lie to a computer it can end up hating you!
 - Nurse 1 keeps changing the BG measurement value until they get an insulin dose that they “like” ... It is lower than what was recommended
 - 3 hours later BG has risen instead of fallen
 - It’s now also Nurse 2’s shift, and the recommended insulin is very high! They give it ...
- A word to the wise for you technologists, ... Never hypo an 850g baby unless you know the reason why (and probably not even then!)
 - We solved this via a key-stroke tracker → raising interesting privacy vs safety issues
- Lesson: Your interface must induce compliance (and track its lack)
 - A very difficult “ask” in NZ-ese

2. The number entry story

- 
- A recent study of ~100k BG measurements entered into a computerised BG protocol showed that 7-8% were in error, of which almost 30-40% of that were potentially non-compliant [Campion et al, 2010, Int Care Med]
 - Analysis showed that ~5% (absolute) would have changed therapy decisions to give more (or less) insulin
 - In part this is a function of the protocol
 - But it's still quite significant
 - **We did our own testing on several basic methods of entering numbers**
 - We too saw 7-9% errors but for some other methods there was up to 22% error
 - Modifying the two lowest methods reduced errors from 7-9% down to less 1-2%.
 - Error trapping of “ridiculous values” is a nice feature!

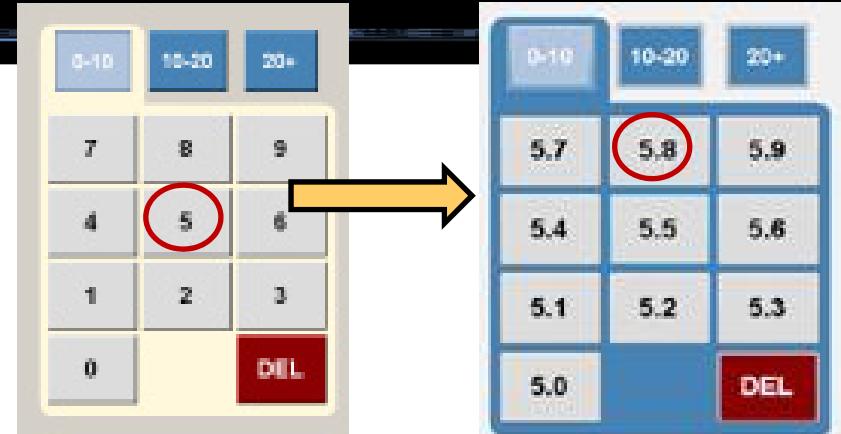
2. The number entry story



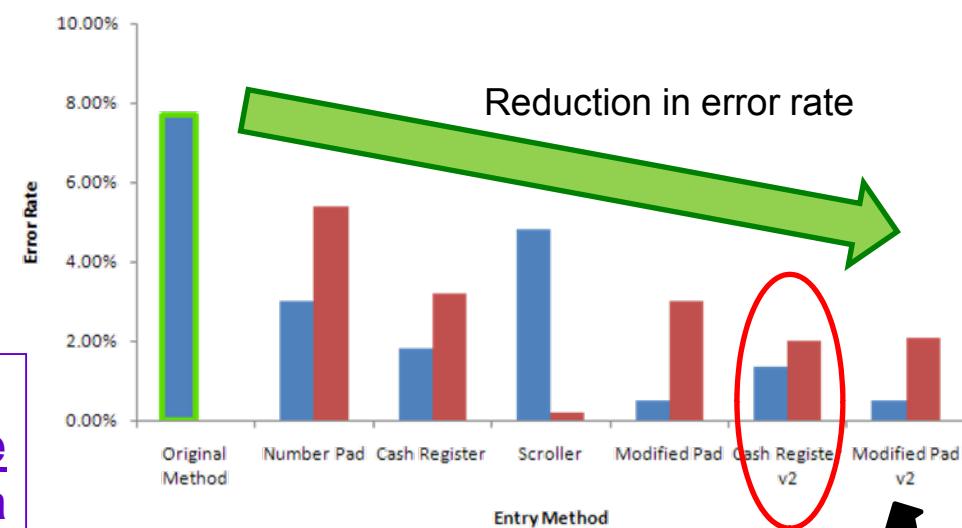
2 taps for both:

- hit “5” and “8”
- hit “5” and “5.8”

Both give 5.8 as input



Error Rate for Different Entry Methods



Best method was equally fast but perceived to be harder so ...

■ Error types:

- Errors entered (blue)
- Errors recognized & corrected (red)

■ Lesson: Number entry is simple (right?!). Simple things can make a big difference in your error rate and thus safety.

3. The ICU SPRINT story

- SPRINT is a successful TGC protocol in use in Christchurch Hospital's ICU for over 5 years
- It gives boluses of insulin for safety and has a very low hypo rate (1-2%)
- 4 years later we came to learn that every patient receives 0.1U/h more insulin than intended or prescribed
- Turning off the infusion rate after a bolus with our particular brand (censored) pumps would necessitate a very long 3-5 minute re-start process (for safety!) every hour
- The Fix = leave the infusion on at its lowest rate of 0.1U/hr
- Lesson: If the technology is hard to use within what you are trying to do, people may well “fix” things without telling you, or it may lead to non-compliance that (quietly) renders your protocol less effective.

4. The (im)patience story

- We recently analysed the Glucontrol protocol, where many BG measurements were taken by blood gas analyser
 - BGAs are very accurate, but take ~5 mins and are typically 1-2 per unit
 - Thus, there were lines every few hours, necessitating waiting
- Actual measurements in each arm of the trial were 30-45% lower than would have been specified by the protocol definition
 - When asked, nurses reported the clinical burden of the protocol was high
 - It affected the tightness of control
- This is not uncommon [Aragon et al, 2006, Amer J. Crit Care; Gurses et al, 2007, Nurs. Res.; ...]
- Lesson: Efficiency counts. Even good technology solutions must work within the clinical constraints of your unit and/or protocol.
 - Here, waiting was like working and not being able to do something else that needed doing.

5. The dosing change story

- Talking about measurement frequency ... How your protocol interacts with those measurements can matter
- SPRINT – specifies an absolute insulin rate. Thus, treatment can catch up with current situation
- Glucontrol – specifies a change in insulin rate. Thus, less measurements equals less opportunities to change insulin rates to catch up with the current metabolic situation, ... leading to greater variability
- Lesson 1: You need to understand about how your protocol and clinical situation can interact (in unintended or unforeseen ways).
- Lesson 2: In French “changez zero” ≠ “changez a’ zero” sound the same... Especially if how you are used to changing insulin is different

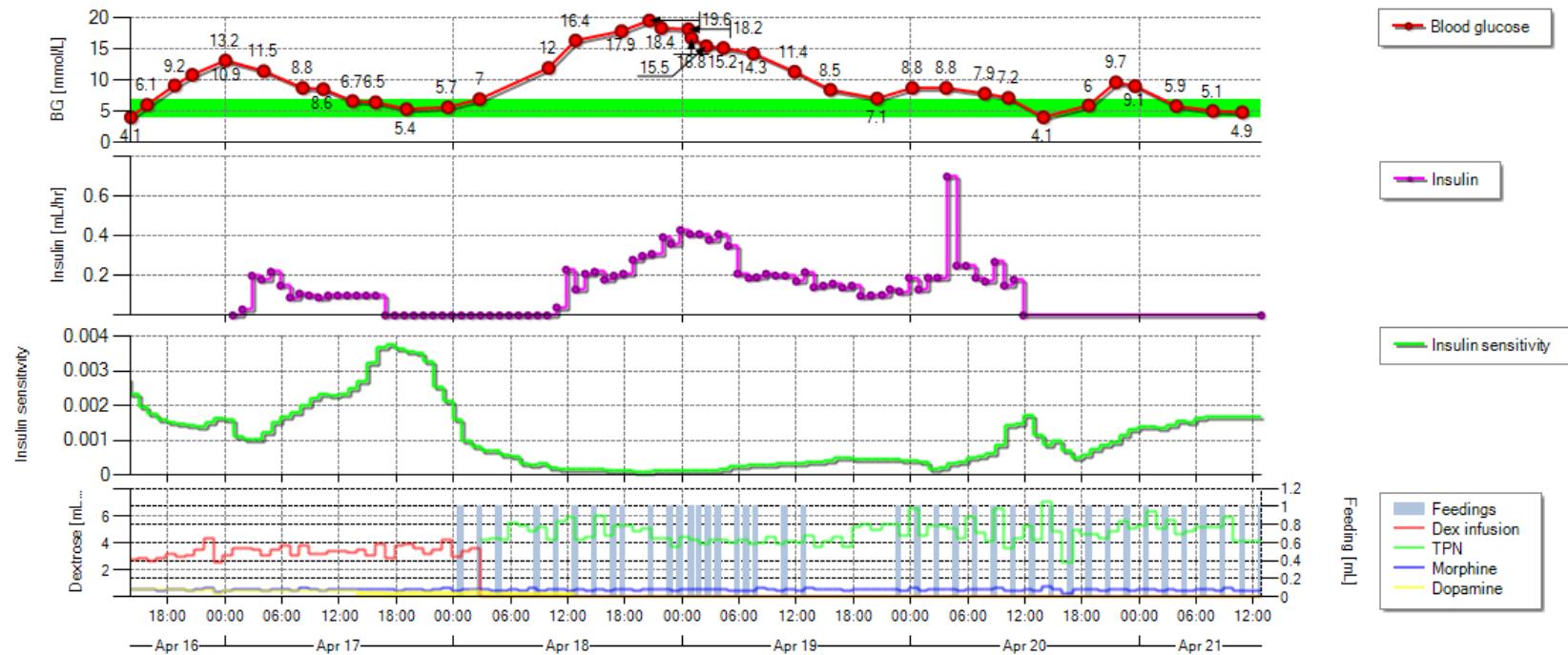
6. The clinical culture story (part 1)

- 
- We've designed GUIs for ICU and NICU. It's the same fundamental problem and thus the same fundamental solutions should work??
 - A long story short version 1.0:
 - **NICU clinical staff** want to double and triple check everything. This necessitates lots of "ok" panels and lots of "tabbed information" one can check
 - Fluid balance is critical and all infusions are re-calculated for good clinical reasons
 - Total administration is tracked "religiously" (yes, an actual quote)
 - **ICU clinical staff** "absolutely hate" this approach (yes, that's a quote).
 - In NZ-ese, the ICU is much "rough and ready" in large part because fluid administration is less urgent clinically
 - **Lesson 1: Clinical culture plays a role ~ know your end-user!**

6. The clinical culture story (part 2)

- A long story short version 2.0:

- **ICU doctors** like lots of graphs and plots to pore over and analyse – **PLANNERS**

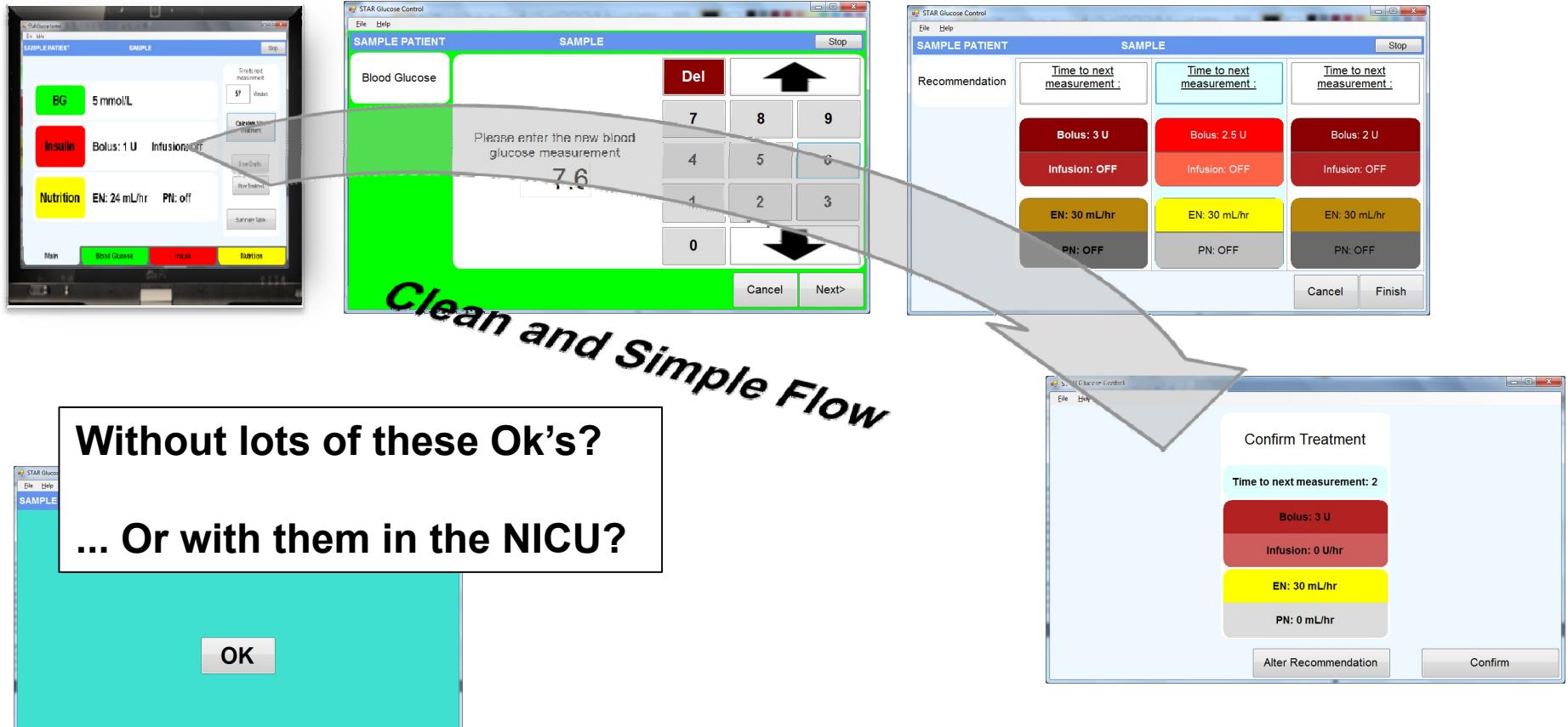


- But ... **ICU nurses** want just the basic data to get on with the job **now** - **GIVERS**

6. The clinical culture story (part 2)

- Lesson 2: Different users (treatment planners and givers) may want very different things from what is effectively the same system

Clean and Simple Flow



The four screenshots illustrate the software's interface from different user perspectives:

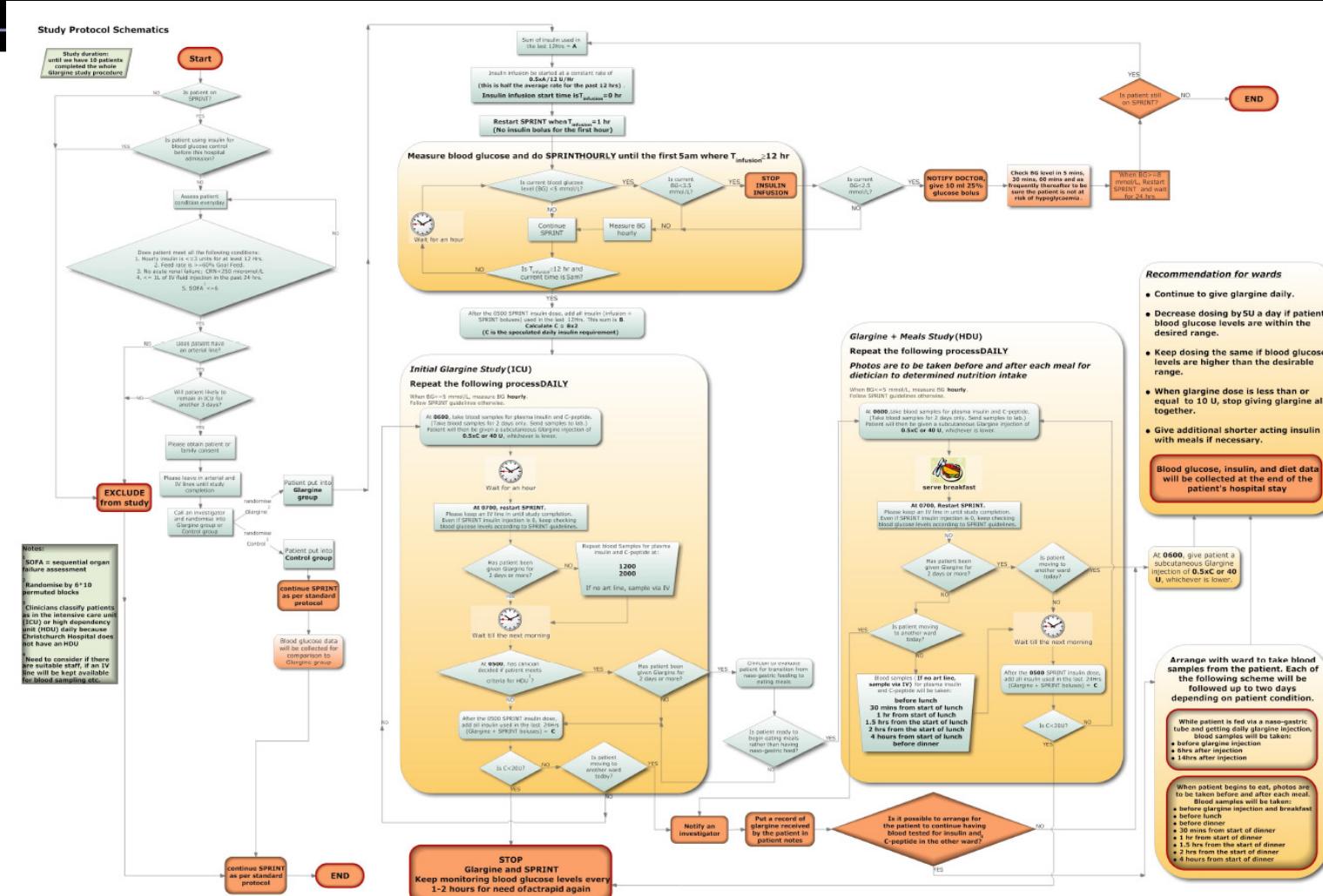
- SAMPLE PATIENT (Top Left):** Shows a main screen with BG: 5 mmol/L, Insulin: Bolus: 1 U Infusion: OFF, and Nutrition: EN: 24 mL/hr PN: off. Buttons include Main, Best Glucose, Next, and Nutrition.
- SAMPLE PATIENT (Top Middle):** A modal dialog box titled "SAMPLE PATIENT" with a sub-section "SAMPLE". It contains a numeric keypad and a message: "Please enter the new blood glucose measurement". The value "7.6" is entered. Buttons include Del, Stop, Up/Down arrows, Cancel, and Next>.
- SAMPLE PATIENT (Top Right):** Shows a grid of treatment recommendations. Rows include: Time to next measurement: (Bolus: 3 U, Infusion: OFF), Time to next measurement: (Bolus: 2.5 U, Infusion: OFF), and Time to next measurement: (Bolus: 2 U, Infusion: OFF). Columns include: EN: 30 mL/hr, PN: OFF, and EN: 30 mL/hr, PN: OFF.
- SAMPLE PATIENT (Bottom Right):** A confirmation dialog box titled "Confirm Treatment" with the message "Time to next measurement: 2". It lists: Bolus: 3 U, Infusion: 0 U/hr, EN: 30 mL/hr, and PN: 0 mL/hr. Buttons include Alter Recommendation and Confirm.

Without lots of these Ok's?

... Or with them in the NICU?

OK

7. The complexity trap story

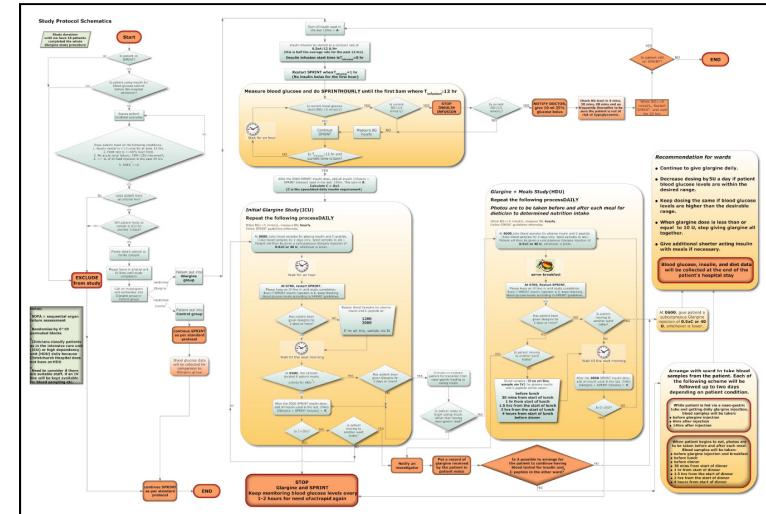


■ This started much simpler, but I think the problem is obvious!

7. The complexity trap story

- Over 4 months of trial design several user groups involved in the trial provided input
- Each wanted to lock down all possible cases and situations
- Each had a different perspective on what was important

- Lesson 1: Avoid complexity
- Lesson 2: Ensure safety and robustness
- Lesson 3: Avoid “feature creep”



- A difficult tradeoff where regulatory requirements can play a similar role in protecting diverse ranges of users

Lessons Learned (?)

- 
- **Keep your interface simple and clean, and design its flow so that users are induced not to “cheat” or take shortcuts that lead to undesirable behaviours**
 - **Design number/data entry to minimise errors and their impact**
 - Alternatively, keep your protocol or system robust to such errors
 - **Make sure your technology is easy to use within the protocol so that no time is lost or wasted (induces better compliance and keeps work low)**
 - **Understand how your protocol will interact with compliance. There is little cure for intervention compliance failures except 2x checking**
 - **Define interface for treatment givers, but accessible for treatment planners, so it's applicable across clinical cultures and users.**
 - **Avoid complexity while ensuring safety, a difficult (no-win?) tradeoff**

She's no worries, mate ...

- It is clear that many of these lessons are tradeoffs or bargains between competing alternatives
- It is thus hopefully equally clear that these issues are critical in the design for (not “and”) implementation of any glycemic control system
- Similarly, if you are coming from the technology development side, then we think you want to (equally) consider how the technology may be used
- Cultural Note: typically said when exactly the opposite should be true!

So ... Finally, those definitions

- 
- **Human Factors:** The physical or cognitive properties, and/or social behaviors that influence the functioning of technological systems.
 - **Human Factors Engineering:** The application of this knowledge to the design of products, processes or services.
 - This includes even “innocuous” interactions from the perspective of the designer
 - **Any technology that is used has an interface (or more than one)**

“Alternative” definitions

- 
- **Human Factors:** The user behaviours or beliefs that can (sometimes) defy rational belief and lead to device/product failures and/or lost revenue or opportunity and/or poor outcome and/or ...
 - **Human Factors Engineering:** The bit that got forgotten in making sure the technology worked in the first place. Plus, it was obvious to us (designers) how it worked!
 - **Lest we forget: What's obvious to you, the clinician, nurse or engineer, is likely not so obvious to others**

Epilogue ... wherein we sum up

- 
- The technology, insulin's / therapeutics, or pumps and sensors don't lead to poor results ... It's how they are/get used!
 - A great protocol is simply average, or worse, if it cannot effectively harness the technology on which it relies.
 - This is all about the potential of **better human factors** to play a critical role
 - Back to the future? Growing standards that lead to consistency across systems and interfaces. (Think PCs 20 years ago and now)

Thank you!

