COSAPIEN INFOGRAPHIC SCOPE

# Scope:

Create a infographic (as described below). Define the json format needed to be able to populate the graphic. In short,

1. combine “arc view” and “donut” to create the reliability view of an individual
2. using zoomable pack layout create the reliability view of a company (by extending donuts to also show groups & adding grey packing to donuts to keep relative arc of reds)

The idea is to follow d3 best practices when this code is created.

To make it IE8 compatible, please use the <https://github.com/shawnbot/aight> library (this is optional only when ie8 detected).

# Intent/Abstract:

The intent of the info graphic is to give the individual an immediate insight into how well the people reporting to him/her are collaborating. Critical here is that the call to action is clear:

in majority of cases the means the work is being worked on (shown in the "live" ring),

specifically work that is OVERDUE, and

to a lesser extent what is due today.

As the user minutes the central infographic a side bar show additional/detailed information pertinent to the selection.

The idea is to have infographic that can be used on tablets and laptops:

the display is broken into two components

a square on the left as main interactive display, and

a rectangular info bar on the right.

Please see the conceptual sketches for:

"individuals to company.jpg" which gives a live view of how reliable individuals/departmants/companies are, and

"relationships.jpg" which gives a live view of how reliable the collaboration relationship between individuals/departmants/companies are

# PHASE 1: Create a prototype of the "individuals to company" view

## PHASE 1.1: (the basic unit = reliability of "a person")

This is a compund circle which contains

the past reliability - "d3-arc%20copy%20v03%20-%20working.html"

if sum of elements are zero, show grey circle

NOTE: handling of zero sized elements

Please innovate the red to yellow, then green transition in steps

(Note: here we wish to show achievement stronger than failure, ie, the green, HOWEVER, "live" is way more important than "past reliability")

surrounded by the live 'reliability' - "live%20donut.html"

{NOTE to Pierre: live view should show active commitments & task requests}

donut width is ratio of reputation radius

which gives outside diameter

shows up to 3 categories (sized according to score)

overdue -> red { fill:FF8800; opacity:1; }

due today -> orange { fill:FF8800; opacity:0.6; }

upcoming -> green { fill:00FF00; opacity:0.3; }

(ideally in same colours as reliability)

if all 3 are zero, then show grey band

? Q: Should there be a small gap between the reliability (ie, psuedo-pie) and live (ie, donut)?

Size of circle should be setable (this will be used in the subsequent )

Reliability score is calculated based on component values (provided in example)

Transparency is used to focus the attention on the overdue and due today live score, ie,

only live overdue and due today is 0% transparent,

upcoming is 15%

all reputation are transparent, eg, 50%

? maybe make green less transparent (to focus on achievements)

(default transparancies could be set)

The suggestion for the JSON is to be roughly:

{"name": "Person", "surname": "Normal", "email": "person@acme.local", "data": [[3,2,1],[93,9,6,3,0,1]]},

where data[0] is live and data[1] is past reliablity

Mouse-over should give a title that explains the infographic, eg,

"Sue Blue <sue@acme.loccal> & team are 97,1% reliable collaborators. They are currently collaborating on 16 tasks, of which 5 are due today, and 9 is already overdue."

"Bob Douglas <bob@acme.local> is 65.1% reliable. Bob is not collaborating with anyone at the moment."

"Thandi Stix <thandi@acme.local> is new to Cosapien. Thandi is currently collaborating on 3 tasks, of which 1 is due today."

For next sub-phase:

Option to show / not show numbers (ie, past reliablity as a percentage and the three live numbers)

Option to affect relative opacity

## PHASE 1.2: reliability of team/company (to be used as tab in dashboard later)

Where the layout is in zoomable hierarchical circles - "pack-hierarchy.html"

where every final circle denoting "a person" (with size determined by sum of live & reliability elements),

and the other circles outside radi is expanded into donuts which contain the sum of the all the containing live scores

Unless if no line reports, then show "a person"

No numbers are shown

Add grey to donut to scale all the values so that the relative arc of red area is consistent

By clicking on a circle

the chosen circle's parent circle is zoomed (or maybe the circle itself -- still to be decided)

and shown more clearly/highlighted as active (ie, it pops)

(In phase 1.3 the info pane on the right is updated to show the selection's basic unit (developed in phase 1) WITH numbers)

{To be done by Pierre: show the associatd task list (through callback)}

Your own reputation is selected by default

Transparency is used to focus the attention on the overdue live score, ie,

only live overdue is 0% transparent,

live due today, and upcoming is, eg, 15% and 30% respectively

unless selected then all live is 0%

all reputation are transparent, eg 75%, unless clicked on then 15%

The extent of the zoom is the circle that encloses the one chosen (ie, zoom to parent)

zoom out is by either clicking on the circle already selected, (or the parent)

Suggestion: the final JSON used is to be calculated in the webpage (but, initially this transformation can be done by hand)

For the individual calculate "size" from the sum of the elements of "data", eg, transforming

{"name": "Person", "surname": "Normal", "email": "person@acme.local", "data": [[3,2,1],[93,9,6,3,0,1]]},

into

{"name": "Person", "surname": "Normal", "email": "person@acme.local", "size":"118", "data": [[3,2,1],[93,9,6,3,0,1]]},

For compound circles

the "name" is calculated from the parent or child individual, eg, consider the individual "name":Sue"

for the parent circle is "name":"Sue & team"

for the child circle is "name":"Sue's reports"

the "size" and "data" is calculated as the sum of all children indivduals (up to infinite depth)

(this algorithm can be simplified by starting counting at child nodes, but that might be more effort than it is worth.