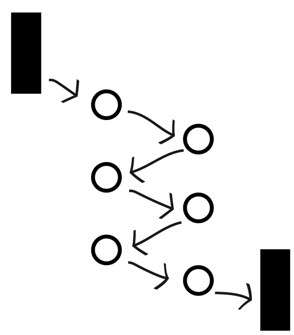
Reorganization of the PC assembly system at the ICL Techno LLC factory to improve performance by 28%

Timur Valeev

The current system is a sequential handover of laptops / PCs along an assembly line, where a team member checks some part of the functionality described in the checklist, or assembles a part of the device.



Analyzing the performance of this system, we introduce the following variables:

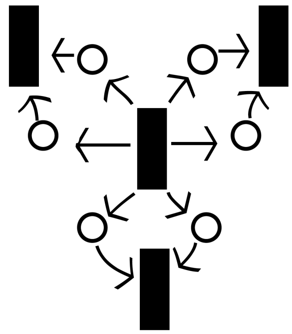
t_h – device handover time

a – time of all necessary operations with the device (without handovers)

N – number of people in the brigade

Then, with a uniform distribution of duties (each spends $\frac{a}{N}$ units of time on checking the functionality / assembling a part), n PCs will be assembled in time t:

$$t(n) = a + t_h(N+1) + (n-1)(\frac{a}{N} + t_h)$$



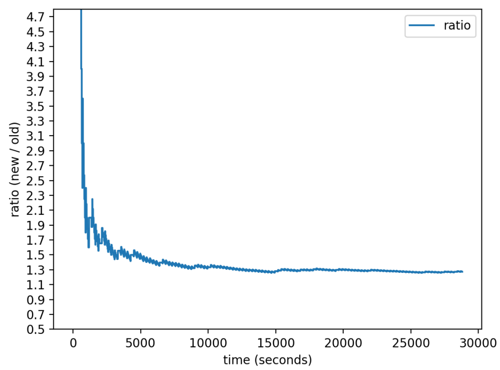
In a system where N people perform operations with the device simultaneously, i.e. each member performs all the operations necessary to complete the check/assembly of one device, n PCs will be assembled in time t:

$$t(n) = \frac{n}{N}(a + 2t_h)$$

With these formulas in hand, we can now proceed with simulations. As one of the main weaknesses of the previous system was temporary delay, resulting from technical difficulties, a 1% chance of a delay whenever a member of a team takes a device was introduced.

```cpp
void System::init(int set_n, int &set_a, int &set_dt) {
        n = set_n;
        a = set_a;
        dt = set_dt;
        finished = 0;

        status.push_back(0);
        status.push_back((a / n) + dt);
        if (n != 1) status.push_back(0);

        for (int i = 3; i <= n; ++i) {
                status.push_back(-1);
        }

        status.push_back(-273);

}

void System::update() {
        for (int i = 1; i <= n; ++i) {
                if (status[i] != 0 && status[i] != -1) {
                        --status[i];
                        if (i == n && status[i] == 0) ++finished;
                }
        }
        for (int i = n; i >= 1; --i) {
                if (status[i] == 0 && status[i - 1] == 0) {
                        double r = ((double) rand() / (RAND_MAX));
                        if (r <= 0.01) {
                                status[i] += a;
                        } else {
                                status[i] = (a / n) + dt;
                                if (status[i + 1] == -1) status[i + 1] = 0;
                        }
                }
        }
}

int System::getFinished() {
        return finished;
}
```

Parameter values have been experimentally derived for testing small laptops with touch screens $a = 232$ c, $t = 4.5$ c
With such parameters ratio of the number of devices tested via the new system to the number of devices tested via the old system over time is:



With $N = 6$, performance of a new system is 28% greater than the performance of the old system. In addition, this system has the following qualities that make it more efficient:

1. The possibility of suspending the work of some team members is excluded, which, in consequence, allows the system to work without significant time losses.

2. An element of friendly competition between members of the brigade (who will do more) is introduced. This aspect is especially well combined with piecework wage.

The disadvantage of the new system is the fact that with an increase in the number of duties per employee, the likelihood of admitting a defect increases, however, this flaw is eliminated by the requirement to mark the operation performed on the checklist immediately after its completion.