7. Raport Totaliztyczny: Sprawa Qt Group

1 Metadane

autor:	**Jacek Marcin Jaworski**
pseudonim:	**Energo Koder Atlant**
pomocnicy autora:	**BRAK**
miejsce:	**Pruszcz Gd.**
utworzono:	**2023-08-21, pon.**
wersja: 497 z dnia:	**2026-06-28**
program składu:	**Libre Office Writer**
sys. op.:	**Kubuntu**
źródło:	**energokod.gda.pl**

Spis treści

2 Wstęp

Qt Group dostarcza wieloplatformową bibliotekę programistyczną Qt dla języka C++. W latach 1991-2008 bibl. Qt w j. C++ zakodowała norweska firma Quasar, później zwana Trolltech. W 2008r. została ona sprzedana fińskiej firmie Nokia. W marcu 2011r. Nokia sprzedała dział Qt innej fińskiej firmie Digia. W paź. 2014r. z firmy Digia wydzielono Qt Company. Rzekomo w 2016r. Digia i Qt Group całkowicie się rozdzieliły, ale jest to nonsens, bo przecież wtedy ktoś musiałby nabyć udziały Qt Company, jednak nabywcy nie ujawniono.

Z [Haavard Nord - s. WWW] wiadomo, że Quasar/Trolltech nigdy nie przynosił zysków. Jednak po opublikowaniu Rap. Tot. pt. „Sprawa Qt Group” Haavard Nord usunął to zdanie. Nie odpowiada też na listy el. z pyt. w tej sprawie.

2.1 Skróty

art.	artykuł
aut.	autor
d.	dzień
dok.	dokument
el.	element
f.	funkcja
j.	język
kat.	katalog
kl.	klasa
kol.	kolejny
l.	liczba
odp.	odpowiedź
ost.	ostatni
poł.	położenie
pow.	powyżej
pub.	publiczny
pyt.	pytanie
Rap. Tot.	Raport Totaliztyczny
roz.	rozdział
s.	strona
sys. op.	system operacyjny
wer.	wersja
wsk.	wskaźnik

Skrótów 4literowych i dłuższych nie tłumaczę, bo uważam je za oczywiste.

2.2 Teza

Bibl. Qt jest najważniejszą powszechnie dostępną bibl. dla j. C++. Qt Group to hamulcowy rozwoju biblioteki Qt. Qt Group robi programistom wodę z mózgu promując j. QML. Dlatego należy rozwidlić bibliotekę Qt by ją oczyścić i naprawić by dalej można było z niej normalnie korzystać w normalnych programach kodowanych w C++.

2.3 Streszczenie

Bibl. Qt zawiera niezbędne kl. do tworzenia prog. w C++. Zawiera też bezużyteczne moduły. W bibl. Qt kl. dostępne z j. C++ są zaniedbane: błędy nie są naprawiane, wygląd i zachowanie kontrolek nie odpowiada dzisiejszym standardom. W Qt prog. i kl. są celowo popsute, tak by nękać i odganiać programistów od C++. Na [Qt Group - s. w Wiki] Qt Group wykazuje spore zyski, jest to zaskakujące, bo aktualny model biznesowy jest b. dziwny i sprzeczny z moją logiką i dostępną mi wiedzą. Analiza w świetle wsk. moralnej popr. daje jednoznacznie negatywny wynik.

3 Mój Oryginalny Wkład

Przyglądałem się Qt i Trolltechowi przez lata i postanowiłem poznać bibl. Qt po jej publikacji na licencji LGPL. Nastąpiło to wkrótce po przejęciu firmy Trolltech przez Nokię w 2008r. Programowanie z użyciem bibl. Qt wymaga używania narzędzi dostarczanych razem z nią: Qt Assistant, Qt Linguistic i Qt Creator. W początkowych latach używania bibl. Qt zgłosiłem kilkadziesiąt błędów. Zgłaszałem też pomysły usprawnień Qt Creatora. Może nie robię jakichś wielkich sys., ale mam komercyjne wdrożenia do tego moje prywatne prog. też działają całkiem przyzwoicie. Mogę się pochwalić, że pracując w gdyńskim Posbit.pl nasz produkt Posnet Pospay Online dostał 1. nagrodę i tytuł „Innowacja Handlu 2019” na ogólnopolskich targach Retailshow [RetailShow - s. WWW].

4 Metoda Badawcza

Opieram się przede wszystkim na swoim ponad 25 letnim (od 1997r.) doświadczeniu w programowaniu w j. C++. W dalszej kol. opieram się na archiwum własnym i na art. w sieci Internet.

5 Fakty

5.1 Czym Jest Bibl. Qt?

Qt to wieloplatformowa bibl. zakodowana w C++. Jest podzielona na kilka mniejszych bibliotek:

- podst. kontrolki graf.;
- kontrolka OpenGL: można wyśw. w oknie Qt scenę OpenGL;
- bazy SQL: łączenie z bazami SQL i wykonywanie zapytań, jednak bez żadnych mechanizmów serializacji danych (kl.->baza i baza->kl.);
- sieć IP;
- port szeregowy;
- geolokalizacja;
- drukowanie;
- dźwięk;
- kl. do parsowania i zapisu plików Json i XML (do plików CSV jest bibl. zewnętrzna [QtCSV - s. WWW]);
- DBus (to w sys. Linuks);
- Przydatnym (choć prostym) uzupełnieniem jest QtTest: do tworzenia testów jednostkowych.

Co dziwne wyodrębniono też moduł QSvg (moim zdaniem powinna to być zwykła wtyczka do renderowania obrazków).

Oprócz tego jest sporo w Qt bibliotek "robiących tłum":

- QtConcurrent: jakiś dziwny potworek do programowania równoległego;
- QtRemote: tajemnicza bibl. do prog. systemów obiektów rozproszonych - z dok. nie można nawet się dowiedzieć w jakim modelu to działa: czy w modelu wywołań synchronicznych czy asynchronicznych;
- QtScript: stary potworek skryptowy z czasów Trolltech zastąpiony w czasach Noki przez potwora QML – od tamtej pory wmawia się nam że do prog. interfejsów dotykowych jest QML i że C++ „się nie nadaje”;
- QWebView: stary silnik wyświetlania plików HTML. Cechował się on dużymi możliwościami i wygodnym API;
- QWebEngine: nowy silnik wyśw. plików HTML. Mimo że powinien on mieć API w 100% kompatybilne z QWebView, to ma on zupełnie inne API – dosłownie wygląda to tak jakby ktoś stwierdził, że QWebView jest zbyt prosty w użyciu.

W przeciwieństwie do QWebView QWebEngine nigdy nie został podczepiony do listy kontrolek Qt Designer. Zrobiono to prawdopodobnie z powodu niestabilności QWebEngine.

5.2 Obecny Stan Prac Nad Qt

Od 2008r, czyli od sprzedaży Trolltech firmie Nokia bibl. Qt praktycznie nie jest rozwijana (innymi słowy proj. jest mrożony). Nie chodzi mi o to że nie dają nam za darmo nowych kl. w C++, chodzi mi o to, że:

- Jest b. mało dostępnych gotowych stylów QSS[1] dla „normalnych” kontrolek dostępnych z C++;
- Nie ma wcale dostępnych gotowych stylów QSS dla twórców prog. w C++ dla ekranów dotykowych;
- Nie wiadomo jak do własnych, nowych kontrolek tworzyć nowe style QSS;
- Zarządzanie oknami w kl. QMdiArea oraz QDockWidget (dostępnych z C++) jest od lat popsute. Należy uznać to za sabotaż Qt Group mający zniechęcać do tworzenia „normalnych” prog.;
- Dom. zach. kontrolek (dostępnych z C++) jest już przestarzałe (niezgodne z intuicją), przez co zawsze wymagają one dodatkowej pracy. Np. kl. QTreeWidget mimo że węzły i liście w tej kontrolce oparte są o kl. QTreeWidgetItem i ma ona 3 stany zaznaczania, to domyślnie nie są one używane. Podczas gdy normalne było by zaznaczanie węzłów jako Qt::PartiallyChecked gdy część liści jest zaznaczona a część nie.

Zamiast tego wszystkiego od 2009r. wśród programistów C++ używających bibl. Qt promuje się potwora QML opartego o język skryptowy Jawa Skrypt.

5.3 Czym Jest QML?

QML jest to zdegenerowany j. skryptowy oparty na Jawa Srypt. QML reklamuje się, jako j. szybkiego tworzenia aplikacji które mogą tworzyć nie tylko zawodowi programiści ale też projektanci. Ale to fikcja. Cechą szczególną QML jest połączenie w jedno deklaracji i implementacji klas oraz tworzenia obiektu – te 3 rzeczy koduje się jednocześnie w tym samym miejscu. Czyli klasę definiuje się w miejscu wystąpienia obiektu w trakcie uruchomienia. TAKIEGO SZALEŃSTWA NIE MA NIGDZIE INDZIEJ!!! Komunikacja między C++ i QML jest możliwa, ale potwornie trudna. Kontrolki QML nie umożliwiają nawet normalnej obsługi kliknięcia.

Pomimo 3 prób komercyjnego użycia QML dla mnie jest on tak nienormalny, że nie jestem w stanie go zrozumieć ani używać.

5.4 Obecny Model Biznesowy Qt Group

- Tak jak wspomniałem we wstępie, Trolltech nigdy nie przynosił zysków właścicielom;

1 Jest to rozw. wzorowane na stylach CSS znanych ze s. HTML. Moim zdaniem QSS to świetny pomysł!

- Jak podano na s. [Qt (software) - s. WWW w Wiki]:

„In 2017, the Qt Company estimated a community of about 1 million developers worldwide[17] in over 70 industries.[18]"

- Wariacki model biznesowy Qt Group polega na udostępnianiu wszystkiego za darmo, a na licencji komercyjnej nie oferowanie niczego więcej (bo kompilator Qt Quick to jakiś żart). Tak wygląda oferta Qt Group dziś w nie. 2023-11-25 patrząc na tabelkę subskrypcji na s. WWW [qt.io/pricing].

- W sklepie [marketplace.qt.io] są bibl. i prog. jakie można dokupić do Qt. Jest to b. dobry pomysł. Jednak przez nie-normalną politykę sprzedaży nie da się tego sklepu używać:

1. Wspierane są WYŁĄCZNIE najnowsze wer. bibl. Qt. Ja kupowałem bibl. QtPdfViewer i mogłem go używać wyłącznie z ost. wer. Qt5 czyli Qt5.15.xx albo z nie w pełni funkcjonalną, wczesną wer. Qt6. Jest to kompletnie nie uzasadnione technicznie, bo dawniej Qt Group chwało się, że ma zasadę, że wszystkie wydania Qt5 są programowo i binarnie kompatybilne - API i ABI są zgodne. Więc nie ma powodów by produkty sprzedawane w [marketplace.qt.io] nie działały na kilkuletnich instalacjach linuksowych distro (które mają starsze wer. bibl. Qt – np. ja dziś w d. 2024-11-17, nie. mam zainstalowane distro Kubuntu 20.04 z Qt 5.12.xx, a ostatnią wer. serii 5 była wer. 5.15.xx);

2. Produkty z tego sklepu można instalować jedynie instalatorem od Qt Group. Tak więc nie można zintegrować tych dodatków z sys. Linuks (razem z którym jest instalowana bibl. Qt). A robi się to istotne w momencie gdy w swoich proj. chcemy bazować na Qt z systemu. Ma to znaczenie w sytuacji gdy prowadzimy własne proj. i modyfikujemy źródła w prog. dostarczanych z distro. Własne proj. łatwo przestawić na bibl. Qt w innym kat. Natomiast przerabianie skryptów budowania prog. dostarczanych w paczkach z distro to masakra. Poza nie może być w sys. Linuks dwóch wer. bibl. Qt (tych samych serii gł., np. wer. 5.12 i wer. 5.15 by się gryzły).

5.5 Rozwiązania Celowo Popsute w Bibl. Qt (Mowa Tu o Tym Co Można Używać z Poziomu C++)

- Nie używanie w bibl. Qt wyjątków. Zamiast nich do sygnalizowania błędów używa się po prostu wart. zwracanej.

ROBI SIĘ TO BY DOPROWADZIĆ PROGRAMISTÓW C++ DO SZALEŃSTWA PRZEZ CIĄGŁE WKLEPYWANIE TEGO SAMEGO KODU DO OBSŁUGI BŁĘDÓW.

Na to jest sposób: dziedziczenie kl. i przykrywanie f. kl. Qt własnymi f., które spr. kod błędu i rzucają wyjątek.

W materiałach propagandowych Qt Group podaje, że w czasach gdy zaczynano prace nad Qt nie było standardu C++, a drugiej strony starano się by Qt działała z jak największą l. kompilatorów (a później dodano, że chcą by Qt pracowała na jak największej l. platform). Ponad to argumentowano, że niektóre z tych kompilatorów miały problemy z prawidłową obsługą wyjątków. Prace nad Qt zaczęto w 1991r. Wtedy już były wyjątki, bo wprowadzono je do C++ w 1990r. Wyjątki objął też pierwszy standard C++ jaki przyjęto w 1998r. No i teraz mamy już 2023r., czyli minęło 25lat, i dalej trzeba klepać w kółko ten sam kod obsługi błędów zwracanych przez wart.!

Dowodem na celowe nie używanie wyjątków w bibl. Qt jest to, że w tym samym standardzie C++ z 1998r. wprowadzono szablony i przestrzenie nazw, które Qt wdrożyło w kodzie bibl.! Więc szablony i przestrzenie nazw można było wdrożyć, a wyjątków nie?

Aby temu zaradzić należy tworzyć własne prywatne i firmowe bibl. separujące ogłupiający kod bibl. Qt od własnych prog. W tych bibl. należy standardowo przezywać wszystkie typy Qt (oraz typy proste C i C++) i w razie konieczności dziedziczyć i rozszerzać je (np. o f. które w razie błędu, zamiast po prostu zwracać kod, rzucają wyjątek).

- QSqlDatabase nie można awaryjnie zamknąć przez zwykłe wywołanie destruktora. W Qt jest zabezpieczenie by nie można po prostu zerwać poł. z bazą danych. A może być to konieczne, np. do przerwania długiego zapytania SQL (np. gdy użytkownik włączy generowanie długiego raportu i zdecyduje się go przerwać po tym

jak się wścieknie długim czekaniem). Blokadę tą uzyskano f. statycznymi (działają one w oparciu o zmienne globalne w pliku C++).

JEST TO ROBIONE PO TO BY UTRUDNIAĆ ŻYCIE PROGRAMISTOM.

Radykalnym obejściem tego problemu jest wydzielenie kodu wywołującego zapytania SQL do osobnego prog. i jego wywoływanie z własnej aplikacji. Wtedy można po prostu ubić prog. by przerwać zapytanie. Wtedy nic złego z bazą się nie dzieje, więc jest to dowód na czysto złośliwe rozw. w kl. QSqlDatabase;

- Wyliczenia w QSizePolicy są podane dokładnie odwrotnie od ich znaczenia.

ROBI SIĘ TO BY DOPROWADZIĆ PROGRAMISTÓW C++ DO SZALEŃSTWA PRZEZ PODWÓJNE MYŚLENIE TYPU: CZARNE JEST BIAŁE, A BIAŁE JEST CZARNE.

Mam takie obejście:

```
#define wMaks QSizePolicy::MinimumExpanding
#define wMin QSizePolicy::Maximum
#define wStala QsizePolicy::Fixed
```

- jw.: Zarządzanie oknami w kl. QMdiArea oraz QDockWidget jest od lat popsute.

NALEŻY UZNAĆ TO ZA SABOTAŻ WOBEC TWÓRCÓW „NORMALNYCH" PROG.

Aby to naprawić konieczne jest rozwidlenie (w j. ang. fork) proj. bibl. Qt i jej naprawa. Dla QDockWidget jest pewna alternatywa w postaci kontrolki KDDockWidgets [KDDockWidgets - s. WWW] szwedzkiej firmy Klarälvdalens Datakonsult AB, ale to tylko ciekawostka z powodu nie korzystnego licencjonowania (za darmo jedynie na licencji GPLv2 lub GPLv3, a licencja komercyjna kosztuje tysiące €).

- jw.: Wygląd i domyślne zachowanie kontrolek jest niezgodne z intuicją i przestarzałe, przez co zawsze wymagają one dodatkowej pracy;

NALEŻY UZNAĆ TO ZA SABOTAŻ WOBEC TWÓRCÓW „NORMALNYCH" PROG.

Aby to naprawić konieczne jest rozwidlenie (w j. ang. fork) proj. bibl. Qt i jej naprawa.

- Z poziomu C++ nie można używać kl. QML. Jest to sztuczne ograniczenie bo te kl. i tak są kodowane w C++. Te kl. QML mają jedną zaletę: są dostosowane do ekranów dotykowych (sprytne tel. i tablety).

  To nachalne wpychanie QML można obejść: standardowe kontrolki Qt można ostylować definiując własne pliki QSS. Jednak wymaga to dodatkowej pracy I TO W DODATKU ARTYSTYCZNEJ!!!. Na 100% jest to wykonalne, bo mam wdrożenia takich prog. (terminale płatnicze Posnet Pospay Online i Posnet Netpay).

NALEŻY UZNAĆ TO ZA SABOTAŻ WOBEC TWÓRCÓW „NORMALNYCH" PROG.

Aby to naprawić konieczne jest rozwidlenie (w j. ang. fork) proj. bibl. Qt i jej popr.

- QApplication::exec() można wywołać jedynie z f. main(). Wywołanie z QApplication::exec() z innej f. kończy się zawieszeniem prog. Aby coś takiego zakodować na prawdę trzeba się postarać! PRAWDOPODOBNIE WYMUSZA SIĘ TO W CELU ANALIZY KRADZIONEGO KODU.

  Aby to naprawić konieczne jest rozwidlenie (w j. ang. fork) proj. bibl. Qt i jej naprawa.

- W okienkowym programie Qt musi istnieć obiekt z oknem głównym. Jak się zamknie okno główne cały program się zamyka. Czyli nawet nie można zrobić kl. LogikaProgramu jaka najpierw otwiera okno konfiguracji, zamyka je, potem otwiera okno gł., zamyka je i na koniec wyświetla okno podsumowania. PRAWDOPODOBNIE WYMUSZA SIĘ TO W CELU ANALIZY KRADZIONEGO KODU.

  Aby to naprawić konieczne jest rozwidlenie (w j. ang. fork) proj. bibl. Qt i jej naprawa.

- Właściwości w kl. dziedziczących po kl. QObject: Te właściwości są tylko po to by można było udostępniać obiekty kl. Qt j. skryptowym. W trakcie prog. w C++ z tych właściwości wcale się nie korzysta. WŁAŚCIWOŚCI W KL. QT SĄ TYLKO PO TO BY MOŻNA BYŁO STWORZYĆ POTWORA QML!

  Aby to naprawić konieczne jest rozwidlenie (w j. ang. fork) proj. bibl. Qt, bo trzeba usunąć wariackie właściwości przydatne jedynie dla twórców portów do j. skryptowych.

- jw.: Wariackie API kl. QWebEngine kl. do renderowania s. HTML;

  NALEŻY UZNAĆ TO ZA SABOTAŻ WOBEC TWÓRCÓW „NORMALNYCH" PROG.

  Aby to naprawić konieczne jest rozwidlenie (w j. ang. fork) proj. bibl. Qt, bo trzeba usunąć wariacki moduł QWebEngine i zaktualizować moduł QWebKit.

- Część f. wirt. nazywa się zdarzeniami np. showEvent, closeEvent, mouseEvent, keyPressEvent. Natomiast prawdziwe zdarzenia nazywane są sygnałami.

ROBI SIĘ TO BY DOPROWADZIĆ PROGRAMISTÓW C++ DO SZALEŃSTWA PRZEZ PODWÓJNE MYŚLENIE TYPU: CZARNE JEST BIAŁE, A BIAŁE JEST CZARNE.

  Tego nie da się naprawić, można jedynie zakodować wszystko od nowa w nowej bibl.

- Standardem w kl. w bibl. Qt jest ukrywanie zmiennych kl. za pomocą wsk. *d do wew. danych kl. To spowalnia dostęp do tych danych i utrudnia życie programistów (uruchamianie ze śledzeniem).

  Aby to naprawić konieczne jest rozwidlenie (w j. ang. fork) proj. bibl. Qt.

- Wew. w bibl. Qt używa się specjalizowanego kontenera QList. Jest on zoptymalizowany do działania na wsk. (ma to znaczenie gdy dane całej kl. to wyłącznie jeden wsk. *d – jw.). Ma taką wadę, że nie nadaje się do pracy w wątkach.

  We własnych prog. nigdy nie należy uzywać QList, tylko zamiast niego QVector.

- Popsuto Qt Assistant: Qt Assistant dawniej miał wygodną obsługę: po wyszukaniu w indeksie i naciśnięcie kl. Enter otwierało nową zakładkę. Było to b. wygodne. Jedynym mankamentem było to, że otwierał te same strony w nowych zakładkach. Teraz Qt Assistant wyszukuje w indeksie ale nie otwiera w kartach tylko zastępuje poprzednią s. dokumentacji. Jest to uciążliwe gdy pracuje się z różnymi kl. Qt. A tak zazwyczaj jest.

  Daje się to naprawić ściągając po prostu źródła do Qt Assistant i je modyfikując.

- W napisach (kl. QString) i w kontenerach (kl. QVector, QList) wiel. (w j. ang. size) Jest określona typem int, czyli l. 32bit. ze znakiem.

  Aby to naprawić konieczne jest rozwidlenie (w j. ang. fork) proj. bibl. Qt, bo w tych kl. należy użyć typu long long czyli l. 64bit. ze znakiem.

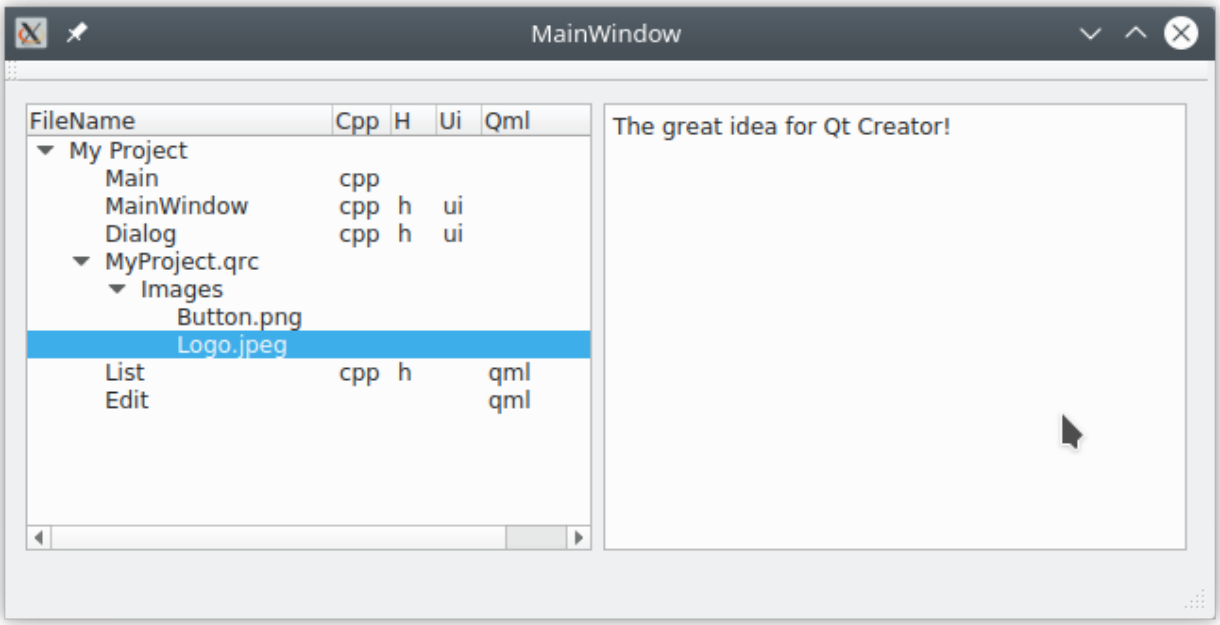
- W QString brak f. zUtf16tLE i z zUtf16BE oraz doUtf16LE i doUtf16BE.

  Być może można te f. jakoś dodać do kl. dziedziczącej po QString.

5.6 Brak Reakcji Na Raportowane Błędy i Na Pomysły Racjonalizatorskie

Ja osobiście w pocz. okresie prog. z użyciem bibl. Qt raportowałem wiele błędów na jakie napotykałem. Jednak przestałem je zgłaszać, bo żadnego z nich nie naprawiono.

Początkowo brakowało IDE do programowania w C++ z Qt, ale po przejęciu Trolltechu przez Nokię stworzyła ona Qt Creator. Jednak miał i nadal ma wady spowalniające pracę i wymagające wielu zbędnych kliknięć. Ja miałem pomysł na jego ulepszenie, by pod Ctrl+Tab zamiast prostej listy plików używać drzewka tak by pliki nagłówkowe i z definicjami f. i z plikami ui występowały w jednej linii. Zrobiłem nawet prototyp by z wizualizować to rozwiązanie. Niestety odp. nie było. Z tamtego czasu zachował mi się zrzut ekranu z takim oknem do przełączania plików:



6 Moralna Ocena Faktów Przytoczonych w Raporcie

6.1 Z Czego Wynikają Kryteria Oceny?

cytat z [Ideologia Geniuszy-Mocarzy]:

„Badania prof. Pająka po godzinach: W toku tych badań naukowych prof. Jan Pająk z NZ ustalił jaka jest "filozofia Wszechświatowego Intelektu" czyli jak Bogowie oceniają ludzkie działanie. Te kryteria to "Wskaźniki Moralnej Poprawności" [totalizm 8p, tom 1, s. 31]. Prof. Pająk podał 8 pierwszych wsk. (w tym roz. są one podane jako pierwsze). Prof. Pająk założył, że Bogowie wyznają filozofię moralności a nie dobra. Jednak w toku moich rozważań nad moralnością okazało się, że jest to nie wystarczające i że Bogowie powinni wyznawać filozofię dobra i moralności by traktować wszystkie istoty po ludzku, bo inaczej będą jak automat do wymierzania kar. Z tego powodu rozszerzyłem wsk. 1. Karma i nazwałem go Sprawiedliwość i Karma uczuciowa oraz ustaliłem kol. 10 Wsk. Moralnej Popr.

Jest prosty sposób na logiczne uzasadnienie konieczności wypełniania poniższych Wsk. Moralnej Popr.: za każdym razem gdy łamie się któryś z tych wsk. pojawia się pasożytnictwo w jednym z jego aspektów: bumelactwie, głupocie lub okrucieństwie.

Aby wykazać, że ktoś jest moralny Wsk. Moralnej Popr. muszą stwierdzać TAK!!! lub BRAK i nigdy NIE! Gdy choć jeden wsk. powie NIE! to znaczy, że ktoś lub coś jest niemoralne.”

6.2 Jakie Wsk. Moralnej Popr. "Są Na Tak" a Jakie "Są Na Nie" i Dlaczego?

Przeanalizujmy po kolei, choć skrótowo, moralną s. proj. Qt Group:

Wsk. Moralnej Popr.	Skrótowy opis Wsk. Moralnej Popr.	Skrótowa analiza zagadnienia	Ocena

1. **Sprawiedliwość i karma uczuciowa!!!**	**Moralne jest przestrzeganie zasady „jaka praca taka płaca”.** **Konieczne jest doświadczenie takich samych uczuć jakie się wzbudziło u innych.** **Moralne jest wzbudzanie u innych takich uczuć jakie sam chciał bym doświadczyć.** **Nie moralne jest robienie innym tego co mi jest niemiłe.**	To że Qt Group udostępnia bibl. Qt za darmo to decyzja udziałowców. Tylko, że to nie uzasadnia nękania programistów z całego Świata. Bo jeśli oni są dłużnikami Qt Group, to nie z własnej woli. Wciskanie popsutych bibl. C++ i ogłupiającego j. QML powoduje negatywne uczucia i silne nerwy, więc jest to łamanie wsk. karma uczuciowa.	**NIE!**
2. **Pole moralne!!!**	**Moralne jest podejmowanie wysiłku intelektualnego (wykazywanie się wysoką inteligencją).** **Niemoralne jest głupie postępowanie.**	Sprzeczne z wsk. pole moralne jest psucie istniejących kl., nie naprawianie zgłaszanych błędów, ignorowanie pomysłów racjonalizatorskich i nie dostosowywanie bibl. Qt do dzisiejszych urządzeń.	**NIE!**
3. **Totaliztyczne dobre uczynki i totaliztyczne grzechy!!!**	**To samo co wsk. Praca Moralna z uproszczeniem, że żyjemy w idealnym Świecie.**	Dostarczanie popsutych bibl. nie jest złym uczynkiem tylko niemoralną pracą.	**BRAK**
4. **Praca Moralna!!!**	**Moralna jest jest dobrowolna praca, najbardziej pod górę pola moralnego i generująca dobrą karmę uczuciową.** **Nie moralna jest praca niewolnicza – jej wyniki są przeklęte.**	Brak prac nad Qt w kontekście C++ to oznaka braku pracy.	**NIE!**
5. **Uczucia!!!**	**Uczucia moralne:** **1. Przyjemne uczucia umysłowe są moralne;** **2. Przyjemności fizyczne są niemoralne;** **3. Dobrowolne pokonywanie przeciwnych uczuć jest moralne;** **Krzywdzenie istot postępujących moralnie jest niemoralne.**	Nie wiadomo co czują pracownicy Qt Group.	**BRAK**

6. Motywacje!!!	**Moralne jest podejmowanie wysiłku i przełamywanie lenistwa.** **Niemoralne jest uleganie lenistwu.** **Niemoralne jest uleganie korupcji w zamian za rezygnację z budowy własnej potęgi.**	Brak prac nad Qt w kontekście C++ to oznaka lenistwa. Chęć zniszczenia cudzego umysłu, to z kolei szataństwo.	**NIE!**
7. Odpowiedzialność!!!	**Moralne jest branie odpowiedzialności za siebie i za innych.** **Niemoralne jest unikanie odpowiedzialności za swoje czyny.**	Sabotowanie całego proj. bibl. Qt, to zupełny brak poczucia odpowiedzialności za siebie, oraz brak poczucia odpowiedzialności za tych którzy jej używają i brak poczucia odpowiedzialności za Cywilizację która ma używać prog. stworzonych z użyciem tej bibl.	**NIE!**
8. Sumienie!!!	**Moralne jest słuchanie sumienia, bo zna ono wszystkie prawa moralne i wydaje nieomylne sądy na temat każdego zrealizowanego działania.** **Niemoralne jest ignorowanie czyli zagłuszanie sumienia.**	Widać, że udziałowcy Qt Group chcący zniszczyć umysły programistom C++ nie mają wcale sumienia (zamiast niego mają „w sercu kamień i mózg z betonu”).	**NIE!**
9. Uczciwość!!!	**Moralna jest uczciwość.** **Wobec wrogów moralny jest podstęp czyli iluzja przyszłości.** **Moralny jest psikus czyli żart chwilowo wprowadzający w błąd i nie powodujący konsekwencji.** **Niemoralne jest kłamstwo na temat przeszłości.**	Cytat z [Mat7:20]:„A więc: poznacie ich po owocach”. No te owoce są popsute, brzydkie, przestarzałe i ogłupiające.	**NIE!**

10. Solidność!!!	**Moralne jest doprowadzanie projektów do końca jaki u odbiorcy nie generuje negatywnych uczuć. Zaleca się by wynik końcowy budził podziw i zachwyt.** **Nie moralne jest „odrabianie pańszczyzny" i wykonywanie zadań „byle jak".**	Tolerowanie błędów w swoich publikacjach jest sprzeczne z wsk. solidność.	**NIE!**
11. Braterstwo!!!	**Moralna jest postawa koleżeńska.** **Nie moralne jest pomijanie lub szykanowanie innych ludzi.** **Moralne jest przestrzeganie zasady „Walczę o człowieka a nie z człowiekiem!".** **Niemoralne jest pogrążanie ludzi przez wrabianie w łamanie jakiegoś tajnego lub niemoralnego prawa.**	Dążenie do denerwowania i nękania ludzi oraz próby niszczenia ich umysłów jest sprzeczne z wsk. braterstwo!	**NIE!**
12. Droga i wdrożenia!!!	**Moralne jest „zrobienie wszystkiego co możliwe" w celu pełnej realizacji proj. Wtedy mówimy, że droga do celu była moralna.** **Niemoralne są brak inicjatywy i zaniechania w wykonywaniu obowiązków, bo to jest sabotaż projektu.**	Wrogie przejęcie jakim było kupienie firmy Trolltech przez Nokię w 2008r. i mrożenie od tamtej proj. bibl. Qt to łamanie wsk. Droga. Jednak na przekór temu podbija się nr wer. bibl. Qt, więc są jakieś lipne wdrożenia.	**NIE!**

13. Sukcesy naukowe i kompetencje!!!	**Moralne jest zdobywanie kol. tytułów naukowo-technicznych, oraz naukowe rozwiązywanie zagadek Naszego Wszechświata oraz inżynierskie rozwiązywanie problemów technicznych.** **Aby legalnie używać danej technologii trzeba na to uczciwie zasłużyć solidnie ucząc się obsługi i uczciwie pracując by legalnie tą technologię wytworzyć lub kupić.**	Z jednej s. wklepano 30.703.306 linii kodu (do wer. Qt 5.12.12 włącznie), a toleruje się w niej liczne błędy. Te błędy w wielu przypadkach uniemożliwiają użycie danej kl., np. QMdiArea i QDockWidget. Wygląda to tak, jakby kompetencje były, ale niewystarczające.	**NIE!**
14. Kultura osobista!!!	**Moralne jest panowanie nad swoim gniewem i nad swoimi żądzami.** **Moralne jest gonienie za legalnie uzyskiwaną wiedzą uzyskiwaną w drodze „białego wywiadu", bo to rozwija intelekt.** **Niemoralne jest szafowanie wyrokami we wściekłości, bo to bezpośrednia droga do niesprawiedliwości i okrucieństwa, czyli zbrodni.** **Niemoralne jest gonienie za kradzionymi informacjami i za kradzionymi technologiami. Oraz:** **Niemoralne jest gonienie za despotyczną władzą, za brudnymi pieniędzmi, za seksem, za alko i narko bo taka gonitwa to kwintesencja pasożytnictwa.**	Nic nie wiadomo o wściekłości akcjonariuszy Qt Group wobec programistów używających bibl. Qt. Nie wiadomo też nic o tym czy gonią za wiedzą.	**BRAK**
15. Kultura techniczna!!!	**Moralne jest wdrażanie pomysłów racjonalizatorskich i tworzenie udoskonalonych narzędzi koniecznych do dalszej pracy.** **Niemoralne jest odrzucanie pomysłów racjonalizatorskich.**	Zakodowanie bibl. Qt wymagało ogromnej wiedzy programistycznej. Jednak tego nie zrobiono w Qt Group tylko w firmach Quasar i Trolltech. A wiem że wielu (najlepszych?) ludzi wygoniono po wrogim przejęciu w 2008r. A od tamtej pory proj. bibl. Qt jest mrożony.	**NIE!**

16. Życie prywatne!!!	**Moralne jest zał. rodziny w celu wychowania 3 lub więcej dzieci, pod warunkiem, że ta rodzina będzie zał. przez mężczyznę ur. jako chłopiec i przez kobietę ur. jako dziewczynka.** **Moralne jest życie towarzyskie oparte na zasadach braterstwa, uczciwości, solidności i zdrowej ekonomi.** **Moralny jest seks na który obie s. wyraziły dobrowolną zgodę w stanie pełnej świadomości/przytomności (bez jakiegokolwiek odurzenia, osłabienia, choroby, bólu, senności ani zmęczenia), oraz pod warunkiem znajomości zaw. Ideologii Geniuszy-Mocarzy, a szczególnie przy dobrym zrozumieniu roz. A może asceza seksualna?** **Moralne jest aktywne spędzanie wolnego czasu.**	Poj. Qt Group nie dotyczy sfery prywatnej tylko zawodowej.	**BRAK**
17. Kultura ekonomiczna i własne utrzymanie!!!	**Moralne jest przestrzeganie w interesach zasady „wygrany-wygrany”.** **Niemoralne jest łamanie praw ekonomii (tak jak to było dawniej w PRL i obecnie w UE).** **Moralna jest praca na własne utrzymanie.**	Firma Qt Group łamie ten wsk. bo dostarczając ogłupiające rozw. dla prog. łamie zasadę „wygrany-wygrany”. To, że wykazuje zyski [Qt Group - s. w Wiki] nie wiele tu zmienia.	**NIE!**

18. Kultura artystyczna!!!	**Moralne jest przestrzeganie zasad tworzenia dzieł sztuki, oraz sztuki użytkowej, które pozwalają na uzyskanie zamierzonego, pozytywnego efektu u odbiorcy (wywołanie Katharsis też jest pozytywne).** **Moralne jest tworzenie maszyn, urządzeń i prog. komp. działających zgodnie z zasadą „zero-conf", czyli prawidłowo wstępnie skonfigurowanych.** **Moralne jest tworzenie maszyn, urządzeń i prog. komp. budzących u odbiorcy zachwyt i inspirację.** **Niemoralne jest tworzenie dzieł, które szkodzą na ciało lub na umysł. Szczególnie niemoralne są chwyty szatańskiej psychologii.**	Oprócz tego, że od lat bibl. Qt ma krytyczne błędy w kluczowych kl., to celowo psute są prog. takie jak Qt Assistant – dawniej on był b. wygodny, a teraz wymaga samodzielnej mod. po to by nie wyrabiać sobie odruchów neurotycznych. Jest to łamanie zasady „zero-conf" i stosowanie szatańskiej psychologii.	**NIE!**

Aby stwierdzić, że ktoś jest moralny wszystkie wsk. moralnej poprawności muszą dać wynik TAK lub BRAK. Występowanie w kol. „Ocena" 14x NIE! i 4x BRAK oznacza, że jest jednomyślność wsk. moralnej popr. Dlatego cały proj. Qt Group jest niemoralny.

6.3 Analiza Znanych Faktów w Świetle Jednego, Wybranego Wsk. Moralnej Popr.

Cytat z [Ideologia Geniuszy-Mocarzy]:

„Miarą wsk. Kultura ekonomiczna jest czysty zysk u wszystkich partnerów w danym proj. biznesowym.

Ten wsk. mówi:

Moralne jest przestrzeganie w interesach zasady „wygrany-wygrany".

[...]

Niemoralne jest łamanie praw ekonomii (tak jak to było dawniej w PRL i obecnie w UE).

Wsk. pochodnym od "ekonomi" jest wsk. "własne utrzymanie":

Moralna jest praca na własne utrzymanie."

1. Współpraca z Qt Group dla firm programistycznych oznacza zawężanie możliwości zamiast ich rozszerzanie. Jest tak bo Qt Group wymusza na programistach C++ przesiadkę na QML czyli zmutowany Jawa Skrypt. Ten QML jest zaprzeczeniem dobrych praktyk programistycznych: w nim w jedno poł. deklarację, definicję i tworzenie obiektu. Takie ogłupiające pomysły są wręcz nie do pomyślenia dla tych co umieją programować w normalnych j. prog.! Programistom C++ celowo blokuje się dostęp do klas QML (które są kodowane w C++) tylko po to by zmuszać programistów do używania tego ogłupiającego j. QML.

2. Qt Group postępuje tak samo jak wszystkie wielkie korpo: tworzy produkty w C++ i klientom karze używać własnościowego j. skryptowego.

3. Trudno powiedzieć jakim cudem Qt Group wykazuje zysk. Bo nawet ich sklep [marketplace.qt.io] jest zaprzeczeniem zdrowego rozsądku (wymusza się użycie jedynie najnowszych wer. bibl. Qt i instalację własnościowym instalatorem Qt Group). Dlatego nie ma mowy by czegokolwiek z tego sklepu używać na sys. Linuks. A sys. Linuks jest dziś (2024-11-17, nie.) najlepszą platformą dostępną dla cywilnych programistów;

4. Nonsensowny marketing (coś jak w Commodore w latach 1985-1994, czyli po wygnaniu założyciela Idka Trzmiela). Zupełny brak obecności w mediach: likwidacja Qt Quarterly (z końcem 2011r.), zero art. sponsorowanych w gazetach i serwisach WWW, zero info o wdrożeniach komercyjnych, zero przykładów proj. wzorcowych. Jedynie na blogu zamieszczają komunikaty typu: „Wyszła nowa wer. Qt X.XX.0, nowości brak.”, albo: „Wyszła nowa wer. Qt Creator XX.0, nowości brak.”.

Pow. fakty jednoznacznie świadczą o łamaniu wsk. moralnej popr. Zwanego Ekonomia i własne utrzymanie w zakresie przestrzegania zasady „wygrany-wygrany”. Negatywnej oceny nie zmieniają wykazywane zyski. Dlatego w świetle tego wsk. proj. Qt Group jest niemoralny.

7 Podsumowanie

Biblioteka Qt od momentu przejęcia norweskiej firmy Trolltech przez fińską firmę Nokia praktycznie nie jest rozwijana. Jest tak na przekór postępów w elektronice w tym np. pojawieniu się sprytnych tel. i tabletów z dotykowymi interfejsami użytkownika.

Aby wspierać nowe interfejsy dotykowe opracowano potwora QML jaki jest zaprzeczeniem filozofii Qt, C++ i w ogóle jest zaprzeczeniem zdrowego rozsądku. Dlatego QML jest kompletnie nie do przyjęcia dla programistów C++.

Model biznesowy Qt Group to fikcja: zarówno oferta subskrypcji, jak zasada działania sklepu [marketplace.qt.io]. Nawet mi nasuwają się różne pomysły jak na tej bibl. można zarabiać. Obecnie już nawet nie wiadomo do kogo należy Qt Company. Qt Group to trojan.

7.1 Rewelacje, Zalety, wadY i partactwA w Proj. Qt Group

7.1.1 Rewelacje

1. Zakodowanie Qt w j. C++ (to zasługa fundatorów);

2. W bibl. Qt zastosowano mechanizm sygnał-slot oparty o prekompilator moc (to zasługa fundatorów).

7.1.2 Zalety

1. W 2023r. bibl. Qt ciągle jest dostępna.

7.1.3 wadY

1. Prog. takie jak Qt Creator i Qt Assistant są uciążliwe w użyciu[2];

2. Zbędne właściwości w kl. dziedziczonych po QObject.

7.1.4 partactwA

1. Krecia robota przeciwko społeczności programistów C++ (np. wspomniane QSizePolicy, albo QApplication::exec(), wymaganie wyświetlania przynajmniej jednego okna gdy prog. używa QApplication);

2. Brak naprawy popsutych kluczowych kl. (takich jak wspomniane QMdiArea i QDockWidget);

2 Bo nie są tworzone zgodnie z zasadą wygody (w j. ang. zero-conf).

3. Nie wszystkie f. pub. są wirt.;
4. Nie wszystkie destruktory są wirt.;
5. Niemożność w kwestii wdrożenia w bibl. Qt wyjątków;
6. Niemożność w kwestii udostępnienia kl. QML na poziomie C++;
7. Brak aktualizacji zachowania kontrolek w bibl. Qt;
8. Brak aktualizacji wyglądu kontrolek na PC (brak nowych stylów QSS);
9. Brak styli QSS dla ekranów dotykowych;
10. Brak możliwości rozszerzania sys. styli QSS (tak by można je było definiować od zera w nowo tworzonych kontrolkach);
11. Niemożność w kwestii integracji zakupionych bibl. w [marketplace.qt.io] z sys. Linuks.
12. Niemożność w kwestii nonsensownego marketingu;
13. Niemożność w kwestii nonsensownego modelu biznesowego.

7.2 Wnioski

Na przekór braku rozwoju Qt od 2008r. do dziś nie ma dostępnego nic porównywalnego do tej bibl. (a przynajmniej ja o niczym takim nie słyszałem). Czy jesteśmy skazani na Qt? W polskich warunkach niestety tak, wynika to z faktu braku środków do napisania porównywalnej bibl. bez wad Qt.

7.2.1 Wątpliwy Jest Sens Przepisywania Qt Od Zera Wynika To z Wad C++

- Braku ABI;

- Brak działania f. wirt. w konstruktorach i destruktorach;

- Brak wybierania najlepszego dopasowania typów, gdy pojawia się jakakolwiek niejednoznaczność (przekleństwo ambigous);

- Brak w kl. aut. tworzenia f. typu virtual w przypadku wszystkich f. w sekcjach public. Jest to konieczne z powodu zapewnienia elastyczności kl. w bibl. Obecnie części pomysłów nie można realizować z tego powodu, że nie wszystkie f. w sekcjach public są typu virtual (np. nie można nadpisać QWidget::update() co oznacza, że nie można swobodnie dodawać kol. el. graf. do istniejących kontrolek, bo w razie zmiany rozmiaru nie ma jak ich uporządkować przed wyświetleniem).

Jednak zdarzają się proste kl. opakowujące takie jak QRect lub QSize lub QPoint w jakich z powodów wydajności nie powinno być pub. f. typu virtual. Dlatego w C++ zwykłe f. pub. powinny być jedynie w strukturach. Wtedy struktura od kl. różniła by się 2 el.:

1. W strukturze domyślną sekcją jest public, a w kl. domyślną sekcją jest private (to już jest w C++);
2. W strukturze domyślną f. w sekcji public nie są typu virtual, a w kl. domyślnie wszystkie f. w sekcji public są typu virtual (to należy dodać do C++).

- Brak znaków Unicode w dekl. makr ani w dekl. f. ani w dekl. kl.

Moim zdaniem w tych 5 kluczowych postulatach niczego się nie traci z obecnego C++, a zyskuje to co gdzie indziej już dawno jest (np. w j. D). Jednak należy zachować podział na pliki deklaracji (*.h++) i pliki definicji (*.c++). By naprawić te wady konieczne jest niewielkie zmodyfikowanie kompilatora C++ i zerwanie kompatybilności z obecnym kodem C++. Dopiero wtedy należało by stworzyć następcę Qt.

7.2.2 Wątpliwy Jest Też Sens Przepisywania Qt Od Zera w j. D Wynika To z Wady j. D

- D ma format plików źródłowych w modelu Asemblera (poł. w jedno deklaracji i definicji f.) a nie w modelu j. C (osobne pliki dla deklaracji i definicji f.). Programowanie w j. D jest procesem odwrotnym do programowania w j. C i C++: w j. D najpierw koduje się definicje f. w kl. i dopiero później można wygenerować plik nagłówkowy z deklaracjami tych kl. Każdy programista C i C++ potwierdzi, że to nienormalny sposób pracy (nawiasem mówiąc tak samo nienormalne jest programowanie sterowane przez testy). Wadą plików źródłowych w modelu j. C/C++ jest puchnięcie kodu o 100% w porównaniu do j. D i Pyton (sprawdziłem to kodując w tych j. przykładowy programik).

7.3 Czy Problem Jest Organizacyjny Czy Jednostkowy?

Problem jest organizacyjny w Qt Group.

7.4 Czy Problem Jest Trwały Czy Czasowy?

Problem jest trwały wypaczenia w firmie Trolltech zaczęły się nasilać od premiery Qt 4.0 w 2005r. (wtedy np. pojawił się Qt Script oraz „modele" kontrolek QListView, QTableView, QTreeView i podobne). Zjawiska negatywne zaczęły galopować po sprzedaży firmy Trolltech firmie Nokia w 2008r. (z j. QML na czele). Po sprzedaży bibl. Qt firmie Digia negatywne zjawiska wcale nie osłabły. Podobnie jest po wydzieleniu Qt Group jako samodzielnej firmy.

7.5 Przewidywania Na Przyszłość

Qt Group najprawdopodobniej dalej będzie cudakować z j. QML. Będą się pojawiać nowe błędy w kl. dostępnych z C++. Qt Group dalej będzie udawać, że marketing nie jest do niczego potrzebny i dalej będzie cudakować ze swoim modelem biznesowym, dalej będzie też cudakować z [marketplace.qt.io]. Wszystko to po to by niszczyć programistów C++ (bo my wiemy, że ten język jest najlepszy do programowania prog., choć nie jest idealny – jw.).

7.6 Zalecenia Na Teraz

Zalecenia dla programistów indywidualnych:

- Nie należy się pakować w QML, bo jest to sprzeczne z ustawą o ochronie zdrowia psychicznego;
- Dalej kodować w C++ i Qt.
- Izolować własny kod C++ od kodu bibl. Qt. W tym celu używane kl. Qt należy przezywać własnymi aliasami używając dyrektyw using i #define. Robimy to po to, by w razie konieczności te aliasy zastępować własnymi klasami (z dodatkowymi zmiennymi i z dodatkowymi f.).

Zalecenia dla polskich firm, oprócz pow. dodatkowo:

- Należy ostylować w QSS kontrolki Qt tak by dało się je używać na sys. z ekranami dotykowymi. Wymaga to też pewnej pracy programistycznej dotyczącej obsługi kliknięć (w Qt kliknięcie jest punktowe a na ekranach dotykowych powinno działać w określonym promieniu). To na 100% działa, bo już wdrożył nieistniejący Posbit.pl (jeszcze dziś, 2023-08-21, można to zobaczyć w terminalach płatniczych Posnet Pospay Online i Posnet Netpay).

7.7 Zalecenia Na Przyszłość

Zalecenie dla społeczności polskich programistów C++:

- Należy rozwidlić[3] proj. Qt oraz poprawić w. wymienione wady Qt;

3 w j. ang. fork

Rozwidlenie Qt nie będzie możliwe do użycia w mikrokontrolerach typu bare-metal (czyli bez sys. op.), bo w nich nie ma bibliotek współdzielonych jakich dystrybucji wymaga licencja LGPL.

Jednak użycie Qt w mikrokontrolerach typu bare-metal i tak jest trochę bez sensu, bo Qt nie jest bibl. czasu rzeczywistego (bo jest oparta o pętlę zdarzeń, która nie gwarantuje kiedy zdarzenie zostanie obsłużone).

- Oferta komercyjnej subskrypcji rozwidlenia Qt powinna być tak atrakcyjna, że uzasadniała by ponoszone koszty. Np.: Na licencji LGPL dostępne były by konieczne narzędzia, ale tylko konsolowe. Natomiast na licencji komercyjnej udostępniane były by wygodne narzędzia[4] z interfejsem graf. (w rodzaju obecnych Qt Creator, Qt Linguist i Qt Assistant. Qt Designer jest zbędny bo używanie tego typu prog. jest sprzeczne z zasadami prof. programowania które wymagają 100% kontroli nad kodem). Nie ma przy tym obawy o konkurencję ze s. obecnych narzędzi graf. od Qt: one też są nienormalne i popsute i wystarczy parę minut poklikać i już jest mnóstwo pomysłów jak je można by naprawić i rozwinąć. Jednak tego typu narzędzia należy zakodować od zera (z powodu zapewnienia sobie do nich pełnych praw autorskich koniecznych do ich sprzedawania);
- Należy stworzyć sklep z dodatkami dla rozwidlonego Qt, tak by udostępniać w nim dodatkowe bibl. i prog. na komercyjnych zasadach. Pomysłów jest mnóstwo: kl. do edycji dok. ODT i ODS, kl. do wyświetlania plików HTML, PDF i innych, wtyczki SQL do popularnych baz danych, bazy no-sql, hurtownie danych, popularne protokoły: HTTP, FTP, CAN, RS. Tego jest i będzie coraz więcej! Tu jest miejsce na biznes!
- Należy dbać o to by społeczność mogła też się włączyć do tego interesu. W tym celu należy zdefiniować proste zasady i jasne kryteria techniczne i biznesowe jakie pozwalają na opublikowanie rozszerzeń dla rozwidlonego Qt.

8 Licencja

Jest to licencja dotycząca tego dokumentu. Pliki wskazywane przez linki mogą być publikowane na innych licencjach. Zasady licencji:

1. **Zezwolenie na kopiowanie** Zezwala się na niekomercyjne kopiowanie tego dokumentu;
2. **Zezwolenie na udostępnianie** Ten dokument można udostępniać (jednak bezpłatnie);
3. **Zabronione modyfikowanie** Tego dokumentu nie można modyfikować ani skracać ani dodawać czegokolwiek.
4. **Ograniczenia licencji nie dotyczą autora.**

9 Bibliografia

Bibliografia

Haavard Nord - s. WWW: Haavard Nord, Painful Lessons From Scaling a Software Company, 2016, https://medium.com/21st-century-decision-making-for-teams/painful-lessons-from-scaling-a-software-company-1d14ee7f9ab3
Qt Group - s. w Wiki: , Qt Group, 2024, https://en.wikipedia.org/wiki/Qt_Group
RetailShow - s. WWW: , Zwycięzcy w konkursie Innowacje Handlu 2019, 2019, https://retailshow.pl/pl/laureaci-2019-2
QtCSV - s. WWW: , GitHub - iamantony/qtcsv: Library for reading and writing csv-files in Qt., 2024., github.com/iamantony/qtcsv
Qt (software) - s. WWW w Wiki: , Qt (software) - Wikipedia, 2024, https://en.m.wikipedia.org/wiki/Qt_(software)
qt.io/pricing: , Pricing and Packaging | Software Stack | Tech Stack | Qt, 2023, qt.io/pricing
marketplace.qt.io: , All products Support | Qt Marketplace, 2024, marketplace.qt.io
KDDockWidgets - s. WWW: , KDDockWidgets, 2024, https://www.kdab.com/development-resources/qt-tools/kddockwidgets
Ideologia Geniuszy-Mocarzy: Jacek Marcin Jaworski, Ideologia Geniuszy-Mocarzy, 2026, https://energokod.gda.pl/monografie/Ideologia%20Geniuszy-Mocarzy.pdf
totalizm 8p, tom 1, s. 31: prof. Jan Pająk, Totalizm, 2004
Mat7:20: św. Mateusz, Biblia Tysiąclecia, 2002

4 w j. ang. zero-conf