

Таким образом, получили формулу для нахождения распределения температурного поля в теле с эллиптическим поперечным сечением бесконечной в виде тригонометрического ряда, содержащего гипергеометрические функции.

В настоящей работе приведено решение распределения температурного поля в теле с эллиптическим поперечным сечением при граничных условиях третьего рода. Решение получено в системе эллиптических координат.

*Список литературы:*

1. Канарейкин, А.И. Распределение температуры в теле эллиптического сечения с внутренним источником тепла при граничных условиях первого рода / А.И. Канарейкин // Вестник Калужского университета, серия: естественные науки. – 2020. – №2. – С. 138.
2. Канарейкин, А.И. Распределение температурного поля в теле с эллиптическим поперечным сечением / А.И. Канарейкин // Научные труды Калужского государственного университета им. К.Э. Циолковского. Сер. Естественные науки. 2016. – Калуга: КГУ им. К.Э. Циолковского, 2016. – С. 250.
3. Гладышев, Ю.А. Об одном методе построения поля температур в цилиндре при наличии распределённых тепловых источников / Ю.А. Гладышев, Е.А. Лошкарева // Научные труды Калужского государственного университета имени К.Э. Циолковского. Материалы региональной университетской научно-практической конференции. Сер. Естественные и технические науки. 2019. – Калуга: КГУ им. К.Э. Циолковского, 2019. – С. 275-279.
4. Гладышев, Ю.А. К вопросу о процессе теплопередачи в многослойном цилиндре при прогревании электрически током / Ю.А. Гладышев, Е.А. Лошкарева, А.И. Быльцов // Научные труды Калужского государственного университета имени К.Э. Циолковского. Материалы региональной университетской научно-практической конференции. Сер. Естественные и технические науки. 2018. – Калуга: КГУ им. К.Э. Циолковского, 2018. – С. 146-152.

Российский государственный геологоразведочный университет имени Серго Орджоникидзе, Москва, РФ

УДК 004.42

DOI: 10.54072/18192173\_2021\_2\_109

**В.Г. Виноградский, А.В. Винокуров**  
**СОЗДАНИЕ МОБИЛЬНОГО ПРИЛОЖЕНИЯ**  
**НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ JAVASCRIPT (REACT NATIVE)**

В статье приведено краткое описание фреймворка «React Native». Описаны способы использования данного фреймворка, различия использования «React» и «React Native». Рассмотрена архитектура «React Native». Рассмотрено два способа создания приложения на «React Native»: через «Expo» и «react-native-cli». Выполнено сравнение этих подходов при создании приложения. Приведен пример создания мобильного приложения для просмотра знаменательных событий Великой Отечественной войны при помощи подхода «Expo».

*Ключевые слова:* React Native; React; Expo; мобильное приложение; JavaScript; Android; iOS.

**V.G. Vinogradsky, A.V. Vinokurov**  
**CREATION OF MOBILE APPLICATION**  
**IN JAVASCRIPT PROGRAMMING LANGUAGE (REACT NATIVE)**

The article provides a brief description of the «React Native» framework. The ways of using this framework, the differences between using «React» and «React Native» are described. The «React Native» architecture is considered. There are considered two ways of creating an application in «React Native»: through «Expo» and «react-native-cli». The comparison of these approaches when creating an application is performed. An example of creating a mobile application for viewing significant events of the Great Patriotic War using the «Expo» approach is given.

*Key words:* React Native; React; Expo; mobile application; JavaScript; Android; iOS.

«React Native» – платформа с открытым исходным кодом для разработки мобильных приложений для iOS и Android [1]. Данная платформа была разработана компанией «Facebook» в 2015 году и на сегодняшний день является одной из самых популярных платформ для разработки мобильных приложений. Среди компаний использующих «React Native», помимо «Facebook», находятся такие компании как «Amazon», «Microsoft», «Tesla» и др.

«React Native» позволяет решать любые задачи, связанные с кроссплатформенностью мобильных приложений. Он построен на классической библиотеке «React.js».

«React.js» – это библиотека JavaScript от «Facebook» для создания интерактивных пользовательских интерфейсов [2]. React позволяет создавать страницы, разделяя их при этом на небольшие фрагменты и компоненты. Несмотря на использование «React» в данной платформе, «React Native»

не использует WebView, т. е. не использует DOM дерево (Document Object Model) и DOM API (Application Program Interface). «React Native» не содержит привычных элементов HTML (Hyper Text Markup Language). Например, «div», «span», «li», «ul» и другие, но содержит в себе так называемые компоненты – нативные модули, которые завернуты в React-компоненты. Существуют как кроссплатформенные компоненты, например: 1) «View»; 2) «Text»; 3) «Image»; так и специфичные для каждой платформы (iOS или Android): 1) «TabBarIOS», «ActionSheetIOS», «DatePickerIOS» и др. (для iOS); 2) «ToolbarAndroid», «BackAndroid», «DatePickerAndroid» и др. (для Android).

Перейти с «React» на «React Native» также достаточно просто. Необходимо простые блоки заменить на «View», строковые элементы на «Text», а списки на «FlatList». В «React Native» для каждого HTML-элемента есть свой подходящий элемент.

В «React Native» нет CSS (Cascading Style Sheets), но поддержка стилей остается, благодаря модулю «StyleSheet». Данный модуль предоставляет метод

«create()», который принимает в качестве аргумента объект CSS. Данный объект является упрощенной моделью CSS, описанное при помощи CSS-объектной нотации.

«React Native» делится на две части:

- Приложение (Нативный код).
- Приложение (JS).

Модуль с нативным кодом содержит в себе нативный UI (User Interface) и OS SDK (набор средств разработки).

Модуль с приложением содержит в себе сам «React Native», а также ядро операционной системы, построенное на движке «Javascript V8», сторонние библиотеки и пакеты, предоставляемые менеджером пакетов (NPM).

Архитектура «React Native» приведена на рисунке 1.

Взаимосвязь между частями (или двумя «bundle») осуществлена при помощи структурного шаблона проектирования «Bridge» (мост), используя JavaScript API.

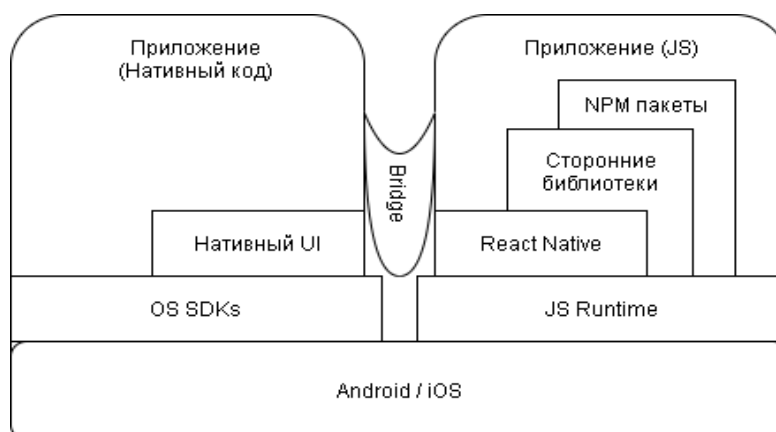


Рисунок 1 – Архитектура «React Native»

Обобщая всё вышеупомянутое, «React Native» имеет следующие преимущества:

- Кроссплатформенность.
- Быстро и дешево.
- Нет прямой необходимости поддержки двух платформ.
- Большое сообщество разработчиков.
- Богатый выбор доступных компонентов и готовых решений.
- Удобный переход от «React» к «React Native».

Создать приложение при помощи «React Native» возможно следующими способами:

- при помощи «Ехро»;
- при помощи «React Native CLI» (Command Line Interface).

«Ехро» – набор инструментов и сервисов, с помощью которых можно развертывать и быстро реализовывать нативные кроссплатформенные мобильные приложения на JavaScript [3].

Данный способ позволяет непосредственно быстро перейти к разработке приложения, но при таком способе в дальнейшем нет возможности добавления модифицированных нативных модулей.

Данный набор инструментов предоставляет API для взаимодействия с различными возможностями мобильного устройства (камера, геолокация, PUSH-нотификации, датчики и другие возможности), а также предоставляет инструмент отладки приложения в режиме реального времени.

При разработке нет необходимости использовать такие IDE (интегрированная среда разработки) как, «XCode» или «Android Studio». «Ехро» позволяет собирать приложение с помощью своих собственных сервисов, а также подписывать приложение и обновлять в режиме реального времени.

«Ехро» функционал полностью зависит от релизов «React Native» и обновить самостоятельно функционал «React Native» внутри «Ехро» не удастся.

В «Ехро» содержится много зависимостей, которые для проекта с большей вероятностью будут не нужны. Эти дополнительные зависимости увеличивают время сборки JS (JavaScript) «bundle», и в релизной сборке появится лишний мусор. Таким образом, размер приложения будет большим, а время запуска приложения увеличится.

«React Native CLI» – это интерфейс командной строки предоставляемый «React Native». «React Native CLI» позволяет добавлять модули в нативную часть приложения, что не позволяет сделать «Expo».

Для сборки приложения или для его подписи необходима установка «XCode» или «Android Studio». Для тестирования приложения необходимо выполнять сборку приложения и тестировать приложение непосредственно на мобильном устройстве.

Каждый из подходов имеет как свои недостатки, так и преимущества. Сравнение двух подходов создания мобильного приложения на «React Native» представлено в Таблице 1.

Таблица 1 – Сравнение Expo и React Native CLI

Особенности	Expo	React Native CLI
Возможность создания нативных модулей	–	+
Размер простейшего приложения	25 мб	5 мб
Наличие Android Studio/XCode	–	+
JS API	+	–
Сторонние библиотеки/ NPM пакеты	–	+
Удобство тестирования	Легче	Сложнее

Выбор способа создания приложения зависит от преследуемой цели при разработке приложения. Если это большое и сложное приложение с большими перспективами и в процессе разработки может появиться необходимость писать нативные модули, то это определено «React Native».

Если же приложение будет достаточно простым, без каких-либо сторонних модулей и библиотек или если приложение будет являться прототипом, то «Expo». На первом этапе при знакомстве с «React Native» вполне достаточно будет Expo версии.

Перед созданием приложения на системе Windows 7 и выше необходимо убедиться в наличии на ПК «Node.js» версии выше 10.

«Node.js» – это среда выполнения JavaScript, созданная на основе движка «Chrome v8 JavaScript» [4].

Для проверки версии «node.js» можно использовать команду:

```
node -v
```

Все команды необходимо выполнять в терминале Windows, например «PowerShell», «Bash» и другие.

Перед установкой «Expo» необходимо убедиться, что установлен менеджер пакетов NPM (Node Package Manager). Чтобы узнать текущую версию NPM, в терминале необходимо ввести команду:

```
npm -v
```

Менеджер пакетов NPM по умолчанию устанавливается вместе с «node.js», но при необходимости можно обновить уже установленную версию до последней версии (опция –g предполагает глобальную установку):

```
npm install npm@latest -g
```

Установка «CLI Expo» (Command Line Interface Expo) выполняется следующей командой:

```
npm install expo -cli -g
```

После установки в терминале будет доступна команда expo, при помощи которой генерируется expo-приложение:

```
expo init my - new - project.
```

Будут предложены следующие варианты:

– «Blank» – минимальное приложение.

– «Blank (Typescript)» – минимальное приложение с поддержкой Typescript.

– «Tabs» – приложение с поддержкой навигации с использованием react-navigation.

После выбора конфигурации приложения будет выполняться установка.

Далее в терминале необходимо перейти в только что созданный проект:

```
cd my - new - project
```

и запустить панель управления «Expo»:

```
expo start
```

После чего, «Expo» создаст локальный сервер для разработки и откроет новое окно для просмотра сервера (Dashboard) (Рис. 2). Также предоставит QR-код для открытия проекта на мобильном устройстве.

Для запуска мобильного приложения на устройстве под системой Android или iOS, необходимо с официального сайта «Expo» (<https://expo.io>) скачать и установить специальное мобильное приложение (Рис. 3), позволяющее взаимодействовать с сервером для разработки. После установки приложения необходимо отсканировать QR-код с сервера разработки и на мобильном устройстве запустится процесс сборки и загрузки JS-bundle приложения (Рис. 4).

Теперь все изменения в проекте будут автоматически запускать процесс сборки приложения, и отображать их на устройстве.

Разрабатывать мобильное приложение можно и при помощи эмуляторов мобильных устройств, таких как «Android Studio Emulator» или «xCode».

Структура проекта мобильного приложения приведена на рисунке 5.

– .expo, .expo-shared – системные папки Expo;

– assets – изображения, иконки, шрифты и др. графические объекты;

– node\_modules – установленные библиотеки, необходимые для инструментов сборки и мобильного приложения;

– .gitignore – файл git-системы для отключения индексации файлов;

– App.js – компонент, являющийся точкой входа в мобильное приложение;

– app.json – конфигурационный файл мобильного приложения;

– babel.config.js – конфигурационный файл babel. Файл необходим для настройки JavaScript транслятора, который позволяет преобразовывать конструкции свежих версий стандарта JS в более поздние;

– package.json – файл, для управления конфигурацией и пакетами приложения в NPM;

– package-lock.json – моментальный снимок всего дерева зависимостей, включающий все пакеты и их установленные версии.

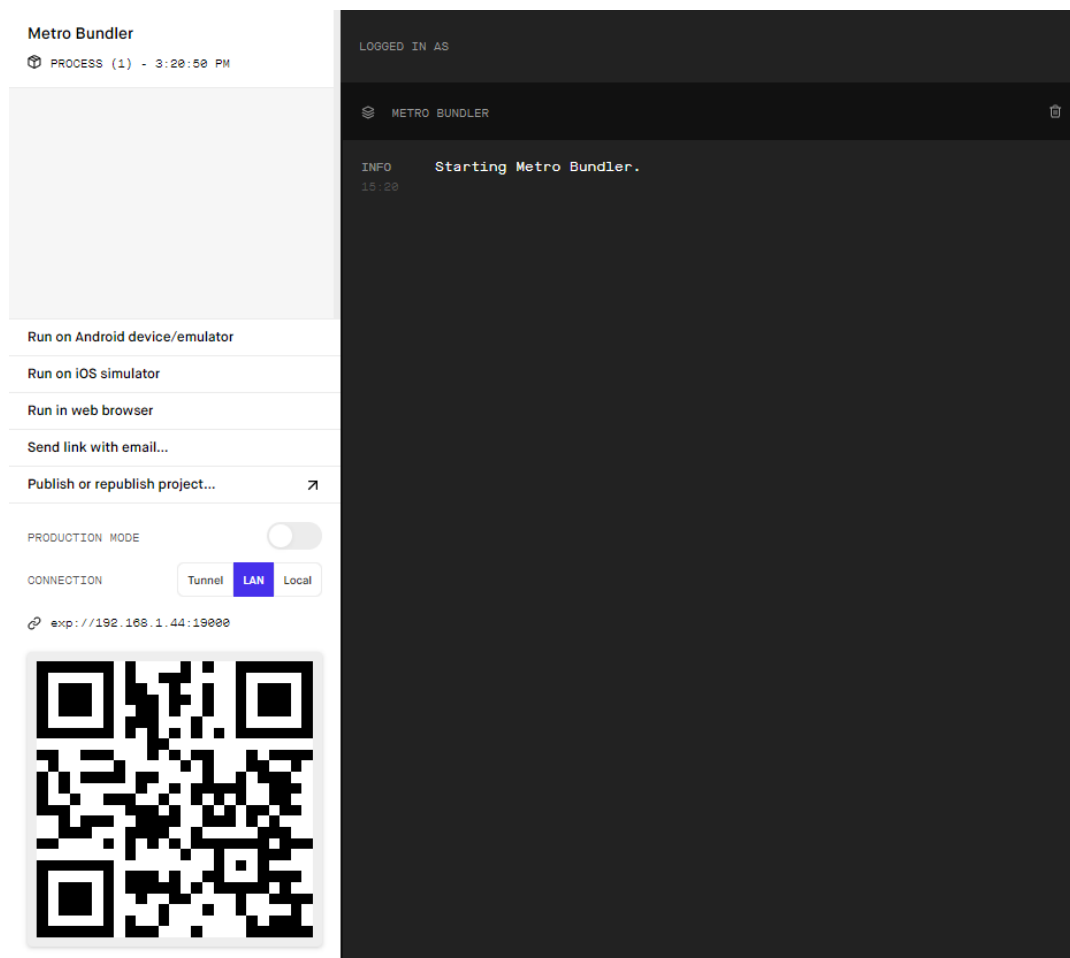
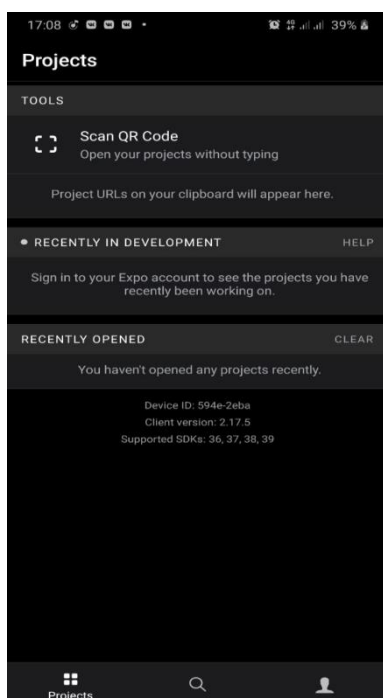
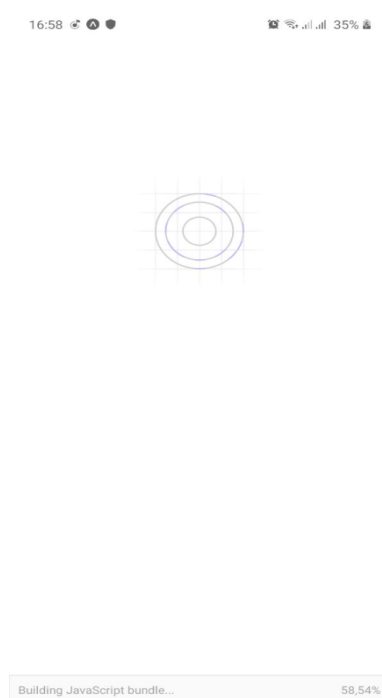


Рисунок 2 – Панель управления Expo

Рисунок 3 – Мобильное приложение  
Expo – платформа для создания приложений  
с использованием JavaScript и ReactРисунок 4 – Сборка и загрузка  
разрабатываемого приложения

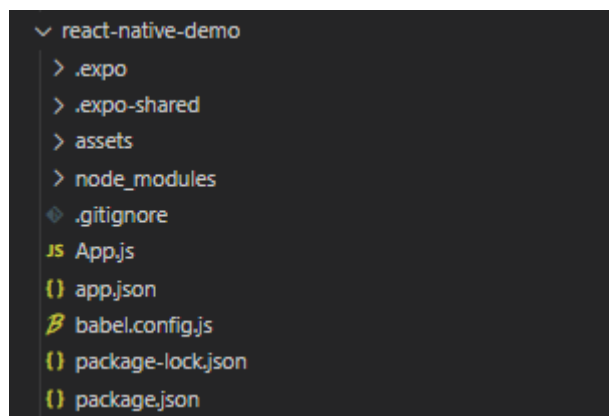


Рисунок 5 – Структура проекта мобильного приложения на основе Expo

Для разрабатываемого мобильного приложения будет использоваться GraphQL-сервер Web-приложения развернутого на <https://калугапамять.рф>. Для взаимодействия с GraphQL сервером будет использоваться пакет «Apollo Client».

Установка «Apollo Client» выполняется командой:  
`npm install @apollo/client graphql`

После установки необходимых пакетов необходимо обернуть приложение в компонент

«ApolloProvider», предоставляемый пакетом «@apollo/client».

Мобильное приложение разрабатывалось при помощи стандартных блоков «View», «Text», «Image», «Button» библиотеки «react-native». На рисунке 5 и 6 представлено разработанное мобильное приложение. Данное приложение позволяет посмотреть текущую памятную дату Великой Отечественной войны.

21:08

94%

04.11.1942

В полосе 16-й армии на отдельных направлениях действуют разведгруппы противника. Снайперами советской армии убито и ранено 20 гитлеровцев. В ночь на 4 ноября 1942 г. в Каменку (Жиздринский район) прибыли немецкие танки. Партизанам удалось подбить два из них. Остальные танки ушли на Рославль. Хвастовичские партизаны, узнав, что немцы направили против них крупное воинское подразделение, организовали засаду. В результате чего каратели потеряли 60 человек убитыми и отступили.



ВЫХОД

Рисунок 5 – Разработанное мобильное приложение. Дата 04.11.1942

Для сборки приложения в формат APK, необходимо в панели управления «Expo» выбрать пункт «Publish or republish project». После чего система предложит авторизоваться или зарегистрироваться. «Expo» поставит проект в очередь на сборку и выдаст ссылки на журнал событий и успешную сборку APK.

21:32

89%

15.11.1942

Снайперами 10-й армии за сутки было уничтожено 64 гитлеровца, стрелками 16-й армии – 13.

ВЫХОД

Рисунок 6 – Разработанное мобильное приложение. Дата 15.11.1942

«React Native» подходит для быстрой разработки кроссплатформенных мобильных приложений. Он подходит для большинства требований современных приложений, а благодаря развитию данной технологии и появлению новых пакетов и библиотек в менеджере пакетов NPM «React Native» существенно облегчает разработку мобильных приложений.

## Список литературы:

1. React Native – A framework for building native apps using React [Электронный ресурс]. – Режим доступа: <https://reactnative.dev/> (дата обращения: 24.06.2020).
2. React – A JavaScript library for building user interfaces [Электронный ресурс]. – Режим доступа: <https://reactjs.org/> (дата обращения: 24.06.2020).
3. Expo [Электронный ресурс]. – Режим доступа: <https://expo.io/> (дата обращения: 24.06.2020).
4. Node.js [Электронный ресурс]. – Режим доступа: <https://nodejs.org/en/> (дата обращения: 25.06.2020).
5. React-native-cli – npm [Электронный ресурс]. Режим доступа: <https://www.npmjs.com/package/react-native-cli/> (дата обращения: 25.06.2020).

Калужский государственный университет им. К.Э. Циолковского, Калуга, РФ

УДК 009

DOI: 10.54072/18192173\_2021\_2\_114

**Н.В. Салтыкова, Н.Г. Ераскин**  
**АНАЛИЗ РЫНКА PLM СИСТЕМ**

Данная статья посвящена анализу рынка основных PLM систем. Приведено краткое описание понятия PLM. Произведен анализ рынка PLM систем в период с 2011 года по 2020 год. Рассмотрены поставщики рынка PLM с наивысшими доходами. Рассмотрены уровни интеграции PLM-систем с приложениями класса CAD/CAPP. Произведен сравнительный анализ функций PLM различных программных продуктов.

*Ключевые слова:* PLM; САПР; cPDM; PDM; NX; CATIA; Solid Edge; Solid Works.

**N.V. Saltykova, N.G. Eraskin**  
**ANALYSIS OF PLM SYSTEM MARKET**

This article is devoted to the analysis of the main PLM systems market. A brief description of the PLM concept is provided. The analysis of the PLM systems market in the period from 2011 to 2020 is made. PLM market suppliers with the highest revenues are considered. The levels of integration of PLM systems with CAD/CAPP applications are considered. A comparative analysis of PLM functions of various software products is made.

*Key words:* PLM; SAPR; cPDM; PDM; NX; CATIA; Solid Edge; Solid Works.

Современные отрасли производственной сферы находятся в условиях конкуренции, вынуждающей их разрабатывать методы уменьшения затрат на разработку и производство новой продукции, а также создавать условия обеспечения её качества. Перед крупными производственными предприятиями стоит задача увеличения скорости выхода продукции на рынок, повышения операционной эффективности, внедрения современных технологий автоматизации основных процессов разработки и выпуска продукции.

На современном этапе развития к традиционным инструментам САПР и автоматизации технологической подготовки производства добавились решения для централизованного управления инженерными данными (Product Data Management, PDM) а также система более широкого функционала управления жизненным циклом изделия (Product Lifecycle Management, PLM).

PLM – информационная основа производственного предприятия на основе методик и технологических решения по выстраиванию бизнес-процессов и интеграции бизнес-систем; рассмотрение изделия в рамках жизненного цикла, начиная от стадии проектирования и подготовки производства, до самого производства, эксплуатации и утилизации. Система PLM обеспечивает управление трехмерными

моделями, чертежами, расчетными моделями, управляющими программами для станков с ЧПУ, технологическими спецификациями, производственными эксплуатационными данными и др.

В 2015 году суммарный объем рынка PLM достиг 38,7 миллиардов долларов, что по сравнению с 2014 годом больше на 2,8% (а в постоянной валюте – на 8,2%). Большинство лидеров рынка показали значительный рост, они продолжили процессы стратегических поглощений, и – в ряде случаев – продолжили вхождение в новые рынки. Аналитики CIMdata считают, что к 2020 году рынок PLM достигнет 52,3 миллиарда долларов при среднегодовом росте 6,3%) [3].

Разнообразие сфер инвестиций и типов решений нашло отражение и в разнообразии поставщиков данных технологий на рынке: в 2015 году шестнадцать провайдеров с доходами, превосходящими 500 миллионов долларов за счёт инструментов и cPDM. Из них три поставщика решений (Dassault Systèmes, PTC и Siemens PLM Software) лидируют как в инструментах, так и в технологии cPDM, а другие (например, Autodesk и Mentor Graphics) в основном фокусируются на поставке инструментов и отдельных сегментах.

Рассмотрим поставщиков рынка PLM с наивысшими доходами на рис. 2